COMMODORE

MENSILE PER UTENTI DI VIC 20 - C64 - C16 - PLUS-4

Lire 3000

Maggio 1985 - Lire 3.000 - Anno II N. 9 - Sped. Abb. Post. Gr. III/70 (CR) - Distr. MePe

systems

COMMODORE

DIRETTORE:
Gloriano Rossi

REDAZIONE/COLLABORATORI:
Eugenio Coppari, Giancarlo De Cobelli, Marco De Martino, Marco De Rosa, Valerio Ferri, Francesco Gatti, Mariangela Guardione, Giulio Marcozzi, Mauro Massetti, Carla Rampi, Ernesto Sidoti, Renzo Zonin.

SEGRETERIA DI REDAZIONE:
Maura Ceccaroli, Piera Perin

UFFICIO GRAFICO:
Mary Benvenuto, Arturo Ciaglia, Paolo Vertuccio

EDIZIONI:
Systems Editoriale S.r.l.
(Registro Nazionale Stampa n. 01500 vol. 15 foglio 793)

DIFFUSIONE e ABBONAMENTI:
Marina Vantini

DIREZIONE, REDAZIONE, PUBBLICITA':
Viale Famagosta, 75 - 20142 Milano
Tel. 02/8467348 - Autorizzazione del Tribunale di Milano N. 103 del 25/2/84
Direttore responsabile:
Agostina Ronchetti

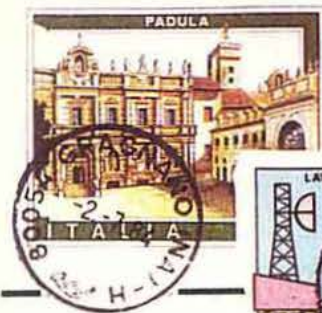
PUBBLICITA':
• Milano: Mirco Croce (coordinatore), Giuseppe Porzani, Michela Prandini, Giorgio Ruffoni, Claudio Tidone, Villa Claudio
Segretaria: Lilliana Degiorgi
• Roma: Spazionuovo
Via P. Foscari, 70 - Tel. 06/8109679

STAMPA:
La Litografica - Busto Arsizio (VA)

Concessionario esclusivo per la diffusione MEPE Spa Via G. Carcano, 32 Milano

Spedizione in abbonamento postale Gruppo III/70

Prezzo della rivista L. 3.000
Arretrati: per richieste fino a 4 numeri L. 5.000 cad., per richieste superiori L. 4.000 cad.
Abbonamento annuo L. 28.000
I versamenti vanno indirizzati a:
Systems Editoriale Srl
V.le Famagosta, 75 - 20142 Milano, mediante assegno bancario, o utilizzando il c/c postale N. 37952207

Per i cambi di indirizzo, indicare, oltre naturalmente il nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500 anche in francobolli.

LA POSTA

● Desiderei sapere come è possibile inserire una riga di BASIC in un programmain linguaggio macchina.

(Ivano Fiumani)

□ *L'unica possibilità è quella di tradurre i codici mnemonici dell'Assembler in istruzioni DATA e poi porre in esecuzione il programma con RUN o GOTO.*

Ciò permetterà alla routine di allocarsi a $C000, 49152 decimale, preservando inoltre la zona scritta in linguaggio BASIC.

Ora ti forniremo un esempio di quanto è stato asserito:

```
10 READ A
20 IFA <> 999 THEN POKE49152+T,A
30 IFA <> 999 THEN T=T+1:GOTO10
40 SYS49152
50 END
60 DATA169,65,141,1,4,96,999
```

Questa breve routine in linguaggio macchina stamperà un picche nell'angolo in alto a sinistra.

● Avrei urgente bisogno di conoscere le varie SYS e POKE !! Consigliatemi un testo.

(Armando Selimi)

□ *Il bisogno di comprendere come funzioni il proprio calcolatore è senza dubbio un'esigenza legittima. Le consiglio quindi di acquistare: 'Il sistema operativo del Commodore 64' edito da EWM.*

● Qual è il significato delle parole poste tra parentesi quadre all'interno dei listati di Commodore?

(Rocchi Andrea)

□ *Le parole poste tra parantesi quadre sono d'aiuto all'utente per saper quale tasto deve digitare. Esaminiamo ad esempio:*

10 PRINT"[DOWN] [RVS] [RVOFF]"

Quando troverai DOWN devi schiacciare il tasto cursore verso il basso. Per quanto concerne RVS e RVOFF premerai rispettivamente CTRL e 9 oppure CTRL e 0.

In ogni numero della rivista Commodore potrai trovare, inoltre, un'elenco completo per l'interpretazione di tutte le parole poste tra parentesi quadre.

● Vi chiedo di pubblicare programmi dedicati ai radioamatori per la gestione della loro attività e delle loro stazioni radio.

(Rosario Vollero)

● Complimenti per la rivista che apprezzo molto. Vi faccio un unico appunto, perchè non pubblicate più programmi per Noi radioamatori.

□ *Cari I8KRV e I4KYO sono oltremodo soddisfatto del vostro interessamento. Vi assicuro che la mia intenzione di aprire una rubrica fissa dedicata ai radioamatori ed appassionati di elttronica vedrà i suoi natali dal prossimo numero con un articolo di un nostro collega sull'SSTV; proseguiremo poi confidando nella vostra collaborazione. Ciao ad entrambi da I2KH Gloriano.*

● Come si usa il registro di stato dell'interruzione del VIC II ?

Come si procede, per esempio, per dividere lo schermo in una parte HI-RES e l'altra in modo carattere normale?

(Bello Massimo)

□ *Questi argomenti saranno oggetto di un prossimo articolo sulla nostra rivista Commodore.*

● Vi sono grato per gli articoli molto interessanti pubblicati in questi primi numeri, desidererei però che affrontaste il linguaggio LOGO con tutte le sue possibili applicazioni nella didattica. Sono interessato, inoltre, a dei programmi inerenti al totocalcio e al totip.

(Belfiore Giuseppe)

□ *Proprio in questi giorni è in edicola una nuova edizione della cassetta Commodore Club, dove è inserito un nuovo tipo di LOGO studiato e realizzato interamente dalla nostra casa editrice. Sarà quindi premura delle testate consorelle: Commodore e Commodore Computer Club di proporre articoli inerenti a questo interessante campo della didattica.*

Per ciò che concerne il discorso del totocalcio e del totip, questi argomenti non possono essere trattati, almeno per il momento, in modo valido su carta.

Inoltre se arrivasse un articolo di un possibile collaboratore che disquisisse sul totocalcio bè ..... ben venga.

● Vorrei sapere se esistono programmi di architettura inerenti l'arredamento di interni.

(Cioffi Giuseppe)

□ *In una delle sue prossime edizioni, la pubblicazione su cassetta Commodore Club presenterà un programma inerente agli argomenti di suo interesse.*

Modello 740/S

La favolosa cassetta che vi permette di eseguire automaticamente il vostro modello 740/S presenta in alcuni casi, dovuti alle diverse Release del Commodore 64, delle difficoltà di caricamento in automatico.

Come ovviare all'inconveniente?

1) riavvolgere la cassetta;

2) digitare LOAD"740S1" e premere il tasto RETURN;

3) date il RUN e verrà caricata l'ultima parte del programma (il tasto PLAY del TAPE deve rimanere premuto).

In questo modo si passa il sistema che noi chiamiamo AUTORUN.

La medesima procedura deve essere eseguita anche per il secondo programma, quello che consente la stampa, effettuando le seguenti operazioni:

1) digitare LOAD"740S/1" e premere il tasto RETURN;

2) date il RUN e verrà caricata l'ultima parte del programma (il tasto PLAY del TAPE deve rimanere premuto).

La Redazione di Commodore Club.

Le stampanti **MT/85, a 80 colonne, e MT/86, a 136 colonne,** rappresentano una nuova frontiera nel settore delle stampanti a basso costo.
Basso costo, ma non bassa qualità e basse prestazioni, infatti ecco le credenziali di questi due nuovi prodotti.
Velocità a 180 cps. bidirezionale ottimizzata, NLQ a 45 cps., grafiche, possibilità di 8 fonti alternative di caratteri e naturalmente la completa compatibilità con il PC IBM.
Il prezzo: il più competitivo del mercato in questa fascia di prestazioni.
Naturalmente anche le MT/85/86 oltre ai trattori hanno anche trascinamento a frizione e consentono pertanto il trattamento del foglio singolo.

MANNESMANN TALLY

20094 Corsico (MI) - Via Cadamosto, 3
Tel. (02) 4502850/855/860/865/870 - Telex 311371 Tally I
00137 Roma - Via I. Del Lungo, 42 - Tel. (06) 8278458
10099 San Mauro (TO) - Via Casale, 308 - Tel. (011) 8225171
40050 Monteveglio (BO) - Via Einstein, 5 - Tel. (051) 832508

adverteam

La Regola di Ruffini

di **Mauro Massetti**

Uno dei più importanti problemi di algebra nei tempi antichi riguardava la risoluzione della scomposizione di polinomi di grado superiore al quinto. E' solo in tempi recenti, infatti, che si è giunti alla definizione di una regola, e la sua formulazione è dovuta a Paolo Ruffini che ha fornito all'algebra moderna la sua ben nota *Regola di Ruffini*.

Dopo che nel Cinquecento gli algebristi italiani avevano trovato le formule risolutive per radicali delle soluzioni algebriche generali sino al quarto grado, si era giunti ad un periodo di stasi nella ricerca: era sorto il problema, che sembrava insolubile, di trovare una formula per risolvere le equazioni di grado superiore al quarto.

Già alla fine del Settecento Lagrange, nella sua opera *Riflessioni sulla risoluzione algebrica delle equazioni*, aveva eseguito un confronto dei vari metodi trovati per la risoluzione delle equazioni di grado uguale o minore al quarto e tentò di spiegare perchè questi metodi non potevano essere estesi alle equazioni di quinto grado. Storicamente il primo autore che pervenne alla dimostrazione dell'impossibilità di risolvere per radicali le equazioni algebriche di grado superiore al quarto, fu appunto il medico italiano Paolo Ruffini.

Prima però di passare alla spiegazione del programma, in che cosa consiste e come viene utilizzata la Regola di Ruffini?

Se si ha un polinomio di grado n che è ordinato secondo le potenze decrescenti di x, si deve dividere per il binomio (x-a) e il quoziente risulta essere un polinomio di grado n-1, i cui coefficienti si trovano utilizzando una particolare metodologia.

Il primo coefficiente è uguale al primo coefficiente del dividendo, e ogni coefficiente successivo viene calcolato moltiplicando il coefficiente precedente per "a" e aggiungendo al prodotto il coefficiente del dividendo che la stessa posizione. Il resto della divisione è dato moltiplicando per "a" l'ultimo coefficiente del quoziente e aggiungendo al prodotto il termine noto del dividendo.

Per meglio chiarire il tutto, ecco un esempio numerico:

Scomporre in fattori applicando la Regola di Ruffini: 3x13-5x12+7x-5=0

Il monomio divisore è (x-1) in quanto per x=1 l'equazione precedente s'annulla. Quindi si passa allo schema.

	3	-5	7	-5
1		3	-2	5
	3	-2	5	0

L'equazione si riduce quindi a:
(x-1)*(3x12-2x+5)

Se il polinomio da scomporre non è completo si dovrà porre nella prima riga dei coefficienti dello schema uno zero al posto dei termini mancanti.

Esempio:
3x14-6x12+4x-32

Il monomio divisore è (x-2) in quanto per x=2 il polinomio si annulla. Si passa quindi allo schema

	3	0	-6	4	-32
2		6	12	12	32
	3	6	6	16	0

che fornisce: (x-2)*(3x13+6x12+6x+16)

E' anche possibile che un polinomio intero in x si possa annullare per x=a,x=b,x=c,.... dove a,b,c.... sono tutti diversi tra loro e quindi questo polinomio risulta divisibile per:
(x-a)*(x-b)*(x-c).......

Esempio: il polinomio 3x13+2x12-7x+2 se scomposto con la Regola di Ruffini risulta essere dato da:
(x-1)*(x+2)*(3x-1)

Il programma

Questo programma comprende anche metodologia di ricerca del divisore. L'unica differenza, che peraltro non limita e non travisa la *Regola di Ruffini*, è che per ragioni di semplicità strutturale delle routines utilizzate e nel caso il coefficiente del primo termine abbia valore assoluto diverso da uno, tutto il polinomio viene semplificato dividendolo per il coefficiente di cui sopra, operazione che non altera in alcun modo, come prima accennato, il polinomio stesso.

La rappresentazione grafica: si è ripetuta quella, a chiunque nota, utilizzata nella fase di scomposizione rendendo immediata la comprensione della metodologia operativa ed una rappresentazione in cascata, sulla videata successiva, dei fattori componenti il polinomio.

A causa della limitata disposizione di spazio su ogni riga (40 colonne) il grado massimo del polinomio processabile risulta essere quattro e l'approssimazione nei risultati è dell'ordine della seconda cifra decimale. Ogni coefficiente può essere costituito al massimo da sei caratteri, segno e virgola compresi.

Qaeste limitazioni non pregiudicano, tuttavia, il carattere didattico del programma, ma anzi lo esaltano sottraendo questo strumento alla tentazione di un suo utilizzo esclusivamente pratico.

Il programma risulta così strutturato.

Alla **1030** vi è il dimensionamento dei vettori e il rinvio alla maschera principale. Dalla linea **1080** alla linea **1200** sono allocate le subroutines di creazione della cornice per le videate e di input dei dati rispettivamente. Dalla **1250** alla **1305** si trovano le istruzioni relative alla maschera di input dei dati.

Dalla **1306** alla **1350** è allocata la routine di elaborazione.

Dalla **1355** alla **1640** si trovano le istruzioni relative alla presentazione dei risultati di scomposizione in polinomi, mentre dalla linea **11330** alla **11400** è allocata la subroutine di scomposizione in fattori. Infine dalla linea **20000** alla **40020** si trovano le subroutines di gestione stringa relative.

```
1000 REM ***************************
1010 REM *   REGOLA DI RUFFINI     *
1020 REM ***************************
1030 DIM IM(5),RI(5),ZZ(5),ZX(5):GO
     TO 1250:REM  INIZIA IL PROGRAM
     MA
1040 REM ***************************
1050 REM * SUBROUTINES CREAZIONE *
1060 REM *CORNICI INTERNA/ESTERNA*
1070 REM ***************************
1080 PRINT"[CLEAR]":FOR H=0 TO 1:FO
     R K=1024+H*960 TO 1063+H*960:P
     OKE K,102:NEXTK:NEXTH
1090 FOR K=1185 TO 1222:POKE K,102:
     NEXTK
1100 FOR K=1064 TO 1944 STEP 40:POK
     E K,102:POKE K+39,102:NEXTK:PR
     INT"[HOME]"
1110 PRINT"[DOWN][11 RIGHT]REGOLA D
     I RUFFINI":RETURN
1120 REM ***************************
1130 REM * SUBROUTINE INPUT DATI *
1140 REM ***************************
1150 X$="":FOR I=1 TO 4:X$(I)="":NE
     XTI:I=1:PRINT"> ";
1160 GET S$:IF S$=CHR$(13) THEN PRI
     NT:FOR J=1 TO I-1:X$=X$+X$(J):
     NEXTJ:RETURN
1170 IF S$=CHR$(45) OR S$=CHR$(46)
     THEN 1190
1180 IF S$<CHR$(48) OR S$>CHR$(57)
     THEN 1160
1190 IF I>4 AND S$<>CHR$(13) THEN 1
     160
1200 X$(I)=S$:PRINTS$;:I=I+1:GOTO 1
     160
1210 REM ***************************
1220 REM * MASCHERA RICHIESTA E    *
1230 REM *  PRESENTAZIONE DATI     *
1240 REM ***************************
1250 GOSUB 1080:FOR I=1 TO 1000:NEX
     TI:FL=0
1260 PRINT"[3 DOWN][2 RIGHT]IMPOSTA
     RE IL GRADO"
1265 PRINT"[2 RIGHT]DEL POLINOMIO (
     (=4) ->          [8 LEFT]";:GOS
     UB 1150:GR=VAL(X$)
1266 IF GR<2 OR GR>4 THEN PRINT"[2
     UP]":GOTO 1265
1270 PRINT"[DOWN][2 RIGHT]FORNIRE I
      COEFFICIENTI"
1275 PRINT"[2 RIGHT]           DI X↑"
     ;GR;" ->          [8 LEFT]";:GO
     SUB 1150:IM(GR)=VAL(X$)
1276 IF IM(GR)=0 THEN PRINT"[2 UP]"
     :GOTO 1275
1280 FOR K=GR-1 TO 1 STEP -1
1290 PRINT"[2 RIGHT]           DI X↑"
     ;K;" -> ";:GOSUB 1150:IM(K)=VA
```

```
    L(X$):NEXTK
1300 PRINT"[2 RIGHT]E IL TERMINE NO
    TO -> ";:GOSUB 1150:IM(0)=VAL(
    X$)
1301 REM *************************
1302 REM *ROUT. ELABORAZIONE DATI*
1303 REM *************************
1305 IF IM(0)=0 THEN PRINT"[2 UP]":
    GOTO 1300
1306 FOR I=1 TO 1000:NEXTI:GOSUB 10
    80
1310 IF ABS(IM(GR))=1 THEN GOTO 132
    5
1320 FOR I=0 TO GR:IM(I)=INT(IM(I)/
    ABS(IM(GR))*100+.5)/100:NEXTI
1325 PT=1:PRINT"[HOME][2 DOWN][4 RI
    GHT]STO' ELABORANDO; ATTENDERE
     PREGO"
1330 GOSUB 11330:GOSUB 20000:GOSUB
    30000:GOSUB 40000:IF FL=1 THEN
     1360
1340 PRINT"[3 DOWN][3 RIGHT]"+A$:PR
    INT"[3 RIGHT]"+H$:PRINT"[3 RIG
    HT]"+B$:PRINT"[3 RIGHT]"+L$:PR
    INT"[3 RIGHT]"+C$
1345 ZX(PT)=ZZ(GR):PT=PT+1
1350 IF GR>2 THEN FOR I=GR TO 1 STE
    P -1:IM(I-1)=RI(I):NEXTI:GR=GR
    -1:PRINT"[5 UP]":GOTO 1330
1351 REM *************************
1352 REM * ROUTINE PRESENTAZIONE *
1353 REM *     DEI RISULTATI     *
1354 REM *************************
1355 PRINT"[HOME][2 DOWN][RIGHT]
          REGOLA DI RUFFINI
       "
1360 GET S$:IF S$="" THEN 1360
1370 GOSUB 1080:PRINT"[4 DOWN][2 RI
    GHT]IL POLINOMIO SI PUO' SCOMP
    ORRE"
1375 PRINT"[2 RIGHT]NEI SEGUENTI FA
    TTORI[2 DOWN]"
1380 FOR I=1 TO PT-1:IF PT-1<1 THEN
     1405
1390 X$=STR$(-ZX(I)):IF LEFT$(X$,1)
    =" " THEN X$="+"+RIGHT$(X$,LEN
    (X$)-1)
1400 PRINT"[3 RIGHT](X"+X$+")"
1405 NEXTI:X$="":IF GR=5 THEN 1620
1410 IF GR=4 THEN 1540
1420 IF GR=3 THEN 1470
1430 Y$="":IF RI(2)=-1 THEN Y$="-"
1440 Y$="("+Y$+"X"
1450 X$=STR$(RI(1)):IF LEFT$(X$,1)=
    " " THEN X$="+"+RIGHT$(X$,LEN(
    X$)-1)
1460 X$=Y$+X$+")":GOTO 1570
1470 Y$="":IF RI(3)=-1 THEN Y$="-"
1480 Y$="("+Y$+"X↑2":U=2:GOSUB 1490
    :GOTO 1530
1490 IF ABS(RI(U))<>1 THEN 1520
1500 X$="+":IF RI(U)=-1 THEN X$="-"
1510 GOTO  1525
1520 X$=STR$(RI(U)):IF LEFT$(X$,1)=
    " " THEN X$="+"+RIGHT$(X$,LEN(
    X$)-1)
1525 RETURN
1530 IF RI(2)=0 THEN 1450
1535 Y$=Y$+X$+"X":GOTO 1450
1540 Y$="":IF RI(4)=-1 THEN Y$="-"
1550 Y$="("+Y$+"X↑3"
1555 U=3:GOSUB 1490:IF RI(3)=0 THEN
     1560
1556 Y$=Y$+X$+"X↑2"
1560 U=2:GOSUB 1490:GOTO 1530
1570 PRINT"[3 RIGHT]"+X$:PRINT"[2 D
    OWN]"
1580 PRINT"[3 RIGHT]VUOI CONTINUARE
     (S/N)"
1590 GET S$:IF S$="" THEN 1590
1600 IF S$="S" THEN 1250
1610 PRINT"[CLEAR]":END
1620 Y$="":IF RI(5)=-1 THEN Y$="-"
1630 Y$="("+Y$+"X↑4":U=4:GOSUB 1490
    :U=3:GOSUB 1490:IF RI(3)=0 THE
    N 1555
1640 Y$=Y$+X$+"X↑3":GOTO 1555
11300 REM ************************
11310 REM *  SUBR. SCOMPOSIZIONE *
11315 REM *      IN FATTORI      *
11320 REM ************************
11330 BA=(-1)*IM(0):RI(GR)=IM(GR)
11340 FOR I=ABS(BA) TO 0 STEP -.005
11350 FOR J=GR-1 TO 0 STEP -1
11355 ZZ(J)=I* SGN(BA)*RI(J+1)
11360 RI(J)=IM(J)+ZZ(J)
11370 NEXTJ:IF ABS(RI(0))>.001 THEN
     NEXTI
11374 S$="[5 DOWN][2 RIGHT]POLINOMIO
     NON SEMPLIFICABILE"
11375 IF ABS(RI(0))>.001 THEN PRINTS
     $:FL=1:GR=GR+1:GOTO 11390
```

```
11376 FOR H=0 TO GR:RI(H)=INT(RI(H)*
      100+.5)/100:ZZ(H)=INT(ZZ(H)*10
      0+.5)/100:NEXTH
11380 ZZ(GR)=INT(I* SGN(BA)*100+.5)/
      100:GOTO 11400
11390 FOR H=1 TO GR:RI(H)=IM(H-1):NE
      XTH
11400 RETURN
11900 REM ************************
11910 REM * SUBROUTINES GESTIONE *
11920 REM *      STRINGA         *
11930 REM ************************
20000 A$="":FOR I=GR TO 1 STEP -1
20010 A$=A$+RIGHT$("      "+STR$(IM(
      I)),6)
20020 NEXTI:A$=RIGHT$(A$,LEN(A$)-1)+
      "|"+RIGHT$("      "+STR$(IM(0))
      ,5)
20030 A$="[5 RIGHT]|"+A$:H$="       |"
20040 FOR I=1 TO LEN(A$)-12:H$=H$+"
      ":NEXTI:H$=H$+"|":RETURN
30000 B$=RIGHT$("          "+STR$(ZZ(GR))
      +"|      ",11)
30010 L$="─────┼":FOR I=GR-1 TO 1 ST
      EP -1
30020 B$=B$+RIGHT$("          "+STR$(ZZ(
      I)),6)
30030 NEXTI:B$=B$+"|"+RIGHT$("      "
      +STR$(ZZ(0)),5)
30040 FOR I=1 TO LEN(A$)-12:L$=L$+"─
      ":NEXTI:L$=L$+"┼─────":RETURN
40000 C$="":FOR I=GR TO 1 STEP -1
40010 C$=C$+RIGHT$("          "+STR$(RI(
      I)),6):NEXTI
40020 C$="[5 RIGHT]|"+RIGHT$(C$,LEN(
      C$)-1)+"|":RETURN
```

Cornucopia

a cura di **Gloriano Rossi**

La rubrica cornucopia è fondata sui contributi che provengono da voi lettori.
Il passaggio di stagione incentiverà l'afflusso in redazione dei vostri piccoli e preziosi programmi?
Nella speranza che ciò si verifichi vi offriamo in questa edizione di cornucopia un'interessante serie di utility.
Nonostante che l'allungarsi delle giornate disincentivi la vostra permanenza dinanzi al computer, non dovete dimenticare il vostro appuntamento mensile con questa rubrica.
Inviate i vostri contributi a:

Spett.Rivista COMMODORE
rubrica Cornucopia
Viale Famagosta 75
20142 Milano

nostra rivista le argomentazioni di un nuovo "genio" della matematica!!

(La Redazione)

```
100 X=5:GOSUB 130:PI=16*SU
110 X=239:GOSUB 130:PI=PI-4*SU
120 PRINTPI:END
130 PO=X:SU=0:SI=1:N=1
140 TE=1/(PO*N):SU=SU+SI*TE
150 N=N+2:SI=-SI:PO=PO*X*X
160 IF TE>1E-9 THEN 140
170 RETURN
```

$ 4F

Cancella video. Quando avete terminato la fase di elaborazione di un programma è necesario curarne adeguatamente anche la veste estetica.

Tramite questa semplice utility potrete cancellare una qualunque delle 25 linee del COMMODORE 64, digitando:
SYS820,Linea da eliminare.
Se volessi eliminare ciò che è contenuto nella riga 23, scriverei:
SYS820,23

Naturalmente si sta facendo riferimento a linee video e non programma.

(ROBERTO MORASSI)

```
100 DATA 32,241,183,76,255,233
110 FOR X=820 TO 825:READ Y:POKE X
    Y:NEXT
```

$ 50

Scrolling alto. Ora vi proporremo delle brevi routine, che vi consentiranno di effettuare degli scrolling in ciascuno dei 4 sensi disponibili.

Il primo programma permette di spostare tutti i caratteri presenti sul video di una posizione verso l'alto.

(La Redazione)

```
100 SC=1024:CO=55296:RO=40
110 FOR K=0*RO TO 24*RO-1
120 POKE SC+K,PEEK(SC+K+RO):POKE C
    O+K,PEEK(CO+K+RO)
130 NEXT
140 FOR K=24*RO TO 25*RO-1:POKE K+
    SC,32:POKE CO+K,6:NEXT
```

$ 51

Scrolling basso. Mediante questa routine sposterete tutti i caratteri di una posizione verso il basso.

Per poter apprezzare questo effetto, è necessario che siano presenti sul video dei simboli grafici sul video.

(La Redazione)

```
100 SC=1024:CO=55296:RO=40
110 FOR K=24*RO-1 TO 0 STEP -1:POK
    E SC+RO+K,PEEK(SC+K)
120 POKE CO+RO+K,PEEK(CO+K):NEXT
130 FOR K=RO*0 TO 1*RO-1:POKE K+SC
    ,32:POKE K+CO,6:NEXT
```

$ 52

Scrolling sinistro. Quando andrà in esecuzione questa routine, tutte le immagini si sposteranno di una locazione verso sinistra.

(La Redazione)

```
100 SC=1024:CO=55296:RO=40:CL=1
110 FOR K=RO*0 TO RO*24 STEP RO:PO
    KE K+SC,32:POKE K+CO,6:NEXT
120 FOR K=RO TO RO*25-1
130 POKE K+SC,PEEK(K+SC+CL):POKE K
    +CO,PEEK(K+CO+CL):NEXT
140 POKE SC+25*RO-1,32:POKE CO+25*
    RO-1,6
```

$ 53

Scrolling destro. Mediante quest'ultimo programma otterremo un effetto opposto rispetto a quello della routine precedente.

(La Redazione)

```
100 SC=1024:CO=55296:RO=40:CL=1
110 FOR K=RO*1-1 TO RO*25-1 STEP R
    O
120 POKE K+SC,32:POKE K+CO,6:NEXT
130 FOR K=RO*25-1 TO RO*0 STEP -1
140 POKE K+SC+CL,PEEK(K+SC)
150 POKE K+CO+CL,PEEK(K+CO):NEXT
160 POKE SC+RO*0,32:POKE CO+RO*0,6
```

$ 54

Conversioni. I computer adottano una logica differente rispetto a quella degli esseri umani.

Per operare correttamente in linguaggi meno evoluti rispetto al Basic, ad esempio l'Assembler, dobbiamo avvalerci del sistema esadecimali.

Un'iniziativa condotta con la nota rivista Computer

PROGRAMMO IN BASIC

Il linguaggio del futuro in un manuale rapido e completo di Clizio Merli
pagg. 224 (L. 9.000)

Il Basic, attualmente il linguaggio più conosciuto - adatto all'utilizzo su qualunque tipo di macchina e in particolare sul personal e gli home-computer - può essere appreso in poche ore con l'ausilio di questo agile manuale.

COME SCEGLIERE UN COMPUTER

Guida pratica per l'acquisto di un mini o di un micro computer professionale di Michele Di Pisa
pagg. 160 (L. 6.000)

Quale modello scegliere tra gli oltre 600 computer commercializzati in Italia? La conoscenza delle caratteristiche delle varie macchine è indispensabile. Con un approccio a "menu" l'Autore vuol essere guida proprio in questa fase.

UTILITY E ROUTINE PER IL COMMODORE 64

di Gloriano Rossi
pagg. 192 (L. 9.000)

L'esecuzione di una istruzione BASIC può richiedere diverse centinaia di passi di programmi in linguaggio macchina. La dimensione dei programmi è ciò che intimidisce maggiormente l'utilizzatore medio di Commodore: aiutato da questo testo chiunque potrà affrontare senza problemi il processo di scrittura di un programma.

BASIC PER LO SPECTRUM

di Maurizio Ariena e Clizio Merli
pagg. 192 (L. 9.000)

Un libro per quanti hanno acquistato il computer ZX Spectrum della Sinclair e intendono sfruttarne appieno tutte le capacità, dall'hardware alla programmazione in assembly (linguaggio macchina).

I volumi, che sono comunque in vendita nelle migliori librerie di tutta Italia, possono anche essere richiesti direttamente all'Editore.
Importante: l'ordine minimo dovrà essere di L. 15.000.

DIZIONARIO DELL'INFORMATICA

Vocabolario Inglese-Italiano di Cultrera, Di Pisa, Giacomelli
pagg. 388 (L. 25.000)

Uno strumento indispensabile per chi si avvicina al mondo dell'informatica e per gli specialisti che hanno l'esigenza di accedere alla dinamica letteratura anglosassone.

Edizioni ACANTHUS

VIALE GRAN SASSO, 23 - 20131 MILANO

Inviatemi i seguenti volumi:

Titolo	quantità	prezzo unitario
spese postali		L. 2.000
	totale	L.

Pagherò contrassegno il dovuto (più L. 2.000 per contributo spese postali) al ricevimento. Potrò restituire i libri entro 8 giorni se non saranno di mio gradimento e avere il rimborso immediato.

COGNOME

NOME

VIA N.

C.A.P. CITTÀ

FIRMA

DATA

Scrivere in stampatello e spedire in busta chiusa.

Questo breve programma effettua le conversioni tra i 2 sistemi precedentemente citati.

E' necessario che il numero fornito in input abbia sempre una lunghezza pari a 4 cifre.

Se volessimo convertire il numero 15, dovremmo digitare quando ci verrà richiesto:
0015.

(La Redazione)

```
100 F2=4096:F3=256:F4=16
110 PRINT"[CLEAR][RVS]CONVERSIONE
    DA ESADECIMALE IN DECIMALE"
120 INPUT Y$:IF LEN(Y$)>4 THEN  RU
    N
130 IF LEN(Y$)<4 THEN PRINT"[UP]AG
    GIUNGI GLI ZERI PER LE CIFRE C
    ON       SIGNIFICATIVE":GOTO 20
    00
140 A$=LEFT$(Y$,1):X$=A$:GOSUB 250
150 A=S*F2
160 B$=MID$(Y$,2,1):X$=B$:GOSUB 25
    0
170 B=S*F3
180 C$=MID$(Y$,3,1):X$=C$:GOSUB 25
    0
190 C=S*F4
200 D$=RIGHT$(Y$,1):X$=D$:GOSUB 25
    0
210 D=S
220 T=A+B+C+D:PRINT"[UP]",T
230 GET Z$:IF Z$="" THEN 230
240 GOTO 120
250 IF X$="A" THEN S=10:RETURN
260 IF X$="B" THEN S=11:RETURN
270 IF X$="C" THEN S=12:RETURN
280 IF X$="D" THEN S=13:RETURN
290 IF X$="E" THEN S=14:RETURN
300 IF X$="F" THEN S=15:RETURN
310 S=VAL(X$)
320 RETURN
```

$ 55

Deviazione standard. Questo programma fornisce la deviazione standard di un determinato numero di elementi scelti dall'utente.

Quando non volete più inserire dei dati digitate:
999999999

(La Redazione)

```
100 PRINT"[CLEAR]":CLR
110 INPUT "VALORE: ";X
120 IF X=999999999 THEN 150
130 T=T+X:S=S+X↑2:N=N+1
140 GOTO 110
150 A=T/N:V=S/N-A↑2:D=SQR(V)
160 PRINT:PRINT:PRINT
170 PRINT"VALORI INSERITI: ";T
180 PRINT"MEDIA: ";A
190 PRINT"VARIANZA: ";V
200 PRINT"DEVIAZIONE STANDARD: ";D
210 PRINT:PRINT:PRINT
220 PRINT"PREMI UN TASTO PER CONTI
    NUARE"
230 GET H$:IF H$="" THEN 230
240 CLR :GOTO 100
```

$56

Directory. Questo programma in linguaggio macchina consente di accrescere il Basic del vostro Commodore 64 mediante il comando DIRECTORY.

Dovete digitare la routine e mandarla in esecuzione tramite il comando RUN.

Se ora scrivete DIR e battete il tasto RETURN, otterrete il menù del dischetto contenuto nel Drive.

```
100 REM    *************************
    ****
110 REM   **
    **
120 REM   **   LETTURA  DIRECTORY
    **
130 REM   **
    **
140 REM   **  DI MARIO VARANONUOVO
    **
150 REM   **
    **
160 REM   **              ROMA
    **
170 REM   **
    **
180 REM   **************************
    ***
190 PRINT"[CLEAR][RVS][9 DOWN][15
    RIGHT]ATTENDERE[RVOFF]"
200 FOR I=40960 TO 49151:POKE I,PE
    EK(I):NEXT
210 FOR I=49152 TO 49239:READ A:PO
    KE I,A:NEXT
220 POKE 41150,68:POKE 41151,73:PO
    KE 41152,210
230 POKE 41189,82:POKE 41190,69:PO
    KE 41191,80:POKE 41192,204
```

```
240 POKE 40988,0:POKE 40989,192
250 POKE 41008,78:POKE 41009,192
260 POKE 1,54:END
270 DATA 234,165,45,133,43,165,46,
    133,44,169,32,141,119,2,169,76
    ,141,120,2
280 DATA 169,111,141,121,2,169,34,
    141,122,2,169,36,141,123,2,169
    ,34,141,124,2
290 DATA 169,44,141,125,2,169,56,1
    41,126,2,169,13,141,127,2,169,
    5,141,128,2
300 DATA 169,76,141,129,2,169,105,
    141,130,2,169,13,141,131,2,169
    ,16,133,198,96
310 DATA 169,1,133,43,169,8,133,44
    ,96
```

$ 57

Ricerca. L'obiettivo di questo programma è effettuare, all'interno di una lista di nomi contenuti nella matrice A$, la ricerca di quello che maggiormente si avvicina al dato fornito in input.

```
100 N=9:DIM A$(N),A(N)
110 A$(1)="GLORIANO ROSSI"
120 A$(2)="ROBERTO SOZZANI"
130 A$(3)="MASSIMO ROSSI"
140 A$(4)="PIPPO PIPPOLINI"
150 A$(5)="PIPPO FILIPPINI"
160 A$(6)="BRUNO BRAZZODURO"
170 A$(7)="FIORENZO"
180 A$(8)="GIOVANNI"
190 A$(9)="GIORGIO"
200 INPUT "QUALE NOME ";X$
210 FOR I=1 TO N
220 IF LEFT$(A$(I),LEN(X$))=X$ THE
    N PRINTA$(I):I=N+1:P=1
230 NEXT
240 IF P=1 THEN 320
250 FOR R=1 TO N
260 FOR I=1 TO LEN(X$)
270 IF MID$(A$(R),I,1)=MID$(X$,I,1
    ) THEN A(R)=A(R)+1
280 NEXTI:NEXTR
290 FOR I=1 TO N
300 IF A(I)>A(X) THEN X=I
310 NEXT:PRINTA$(X)
320 END
```

$ 58

Pallina. Una pallina si muoverà rapidamente sul vostro video, seguendo un percorso irregolare.

```
100 PRINT"[CLEAR][2 DOWN]"
110 A=INT(RND(1)*15):B=INT(RND(1)*
    15):PRINT"[LEFT]  ";
120 FOR K=1 TO A:PRINT"[LEFT] [DOW
    N]●";:NEXT:FOR Z=1 TO 10:NEXT
130 FOR K=1 TO B:PRINT"[LEFT] [DOW
    N][2 LEFT]●";:NEXT:FOR Z=1 TO
    10:NEXT
140 FOR K=1 TO A:PRINT"[LEFT] [UP]
    [2 LEFT]●";:NEXT:FOR Z=1 TO 10
    :NEXT
150 FOR K=1 TO B:PRINT"[LEFT] [UP]
    ●";:NEXT:FOR Z=1 TO 10:NEXT
160 FOR I=1 TO 10:NEXT:GOTO 110
```

$59

Sinusoide. Digitando questo listato potrete apprezzare il movimento sinusoidale di una sprite.

```
 5 REM  G.B. SOFTWARE 1985
 6 PRINTCHR$(147)
10 FOR K=0 TO 62:POKE 832+K,255:N
   EXT
20 V=53248
30 POKE 2040,13:POKE V+39,1:POKE
   53280,0:POKE 53281,0
40 POKE V+21,1:K=1    :K1=50
50 POKE V,100:POKE V+1,100
60 FOR X=20 TO 255 STEP .8
70 POKE V,X
80 Y=K1-SIN(X/5)*30
90 POKE V+1,Y:NEXT
00 K=K+1:K1=K1+10:POKE V+39,K:GOT
   O 50
```

$ 5A

Basta errori. Chi elabora un programma, spesso non si preoccupa delle difficoltà che incontreranno le persone che cercheranno di digitarlo.

L'utility che vi proponiamo sopperisce a questo genere di difficoltà.

Dopo aver caricato questo routine digitate RUN e iniziate la vostra normale fase di programmazione.

Al termine di ogni linea vi verrà fornito un numero nell'angolo in alto a sinistra.

Ponetelo sotto forma di REM alla fine della linea digitata.

Se ripeteremo questa operazione al termine di ogni riga, faciliteremo il compito di colui che dovrà digitare il programma.

$ 4B

Rallenta List. Osservare attentamente il listato di un programma, sul Commodore 64, può risultare assai problematico.

Tramite il tasto CTRL (Control) è possibile rallentare il flusso delle righe di un programma, ma difficilmente saremo in grado di valutarne attentamente il loro contenuto.

Mediante questa utility potrete, dopo aver dato il comando RUN, fermare il listafto mediante il tasto funzione F1 o rallentarlo tramite F3.

Questa routine è scritta in linguaggio macchina e il caricamento di un successivo programma non la potrà assolutamente cancellare.

(La Redazione)

```
100 POKE 251,230
110 FOR D=0 TO 22:READ M:POKE 4915
    2+D,M:NEXT
120 POKE 774,0:POKE 775,192
130 DATA 72,165,197,201,4,240,250,
    201,5
140 DATA 208,8,165,251,133,162,165
    ,162,48,252,104,76,26,167
```

$ 4C

Merge 64. Il BASIC del Commodore 64 non contempla l'istruzione MERGE.

Questa routine consente di unire due programmi, di cui uno residente nella memoria del calcolatore e l'altro situato sul nastro del registratore.

Dopo aver digitato la routine, ponetela in esecuzione mediante il comando RUN.

Ora caricate nella memoria del Commodore 64 il software che maggiormente vi aggrada e quindi battete la seguente linea:

SYS40705"nome del programma residente sul registratore".

Effettuata questa operazione, comparirà sul video il messaggio PRESS PLAY ON TAPE, premete il tasto richiesto del registratore e quando sarà terminato il caricamento i 2 programmi risulteranno unificati.

Vi raccomandiamo naturalmente di verificare che i 2 listati abbiano una differente rinumerazione.

La routine di MERGE è scritta in linguaggio macchina ed è allocata in una zona di memoria che non dovrebbe creare problemi di coesistenza con altri programmi.

(La Redazione)

```
100 PRINT"[CLEAR][BIANCO]":POKE 53
    281,0:POKE 53280,0
110 POKE 55,0:POKE 56,159:CLR :S=4
    0705:FOR J=S TO S+78:READ V:PO
    KE J,V:NEXT:NEW
120 DATA 169,0,133,10,32,212,225,1
    65,43,72,165,44,72,56,165,45,2
    33,2,133,43,165
130 DATA 46,233,0,133,44,169,0,133
    ,185,166,43,164,44,169,0,32,21
    3,255,176,14,134
140 DATA 45,132,46,32,51,165,104,1
    33,44,104,133,43,96,170,201,4,
    144,244,240,10
150 DATA 104,133,44,104,133,43,24,
    108,0,3,164,186,136,240,209,20
    8,239
160 BLOCKS FREE.
```

$ 4D

Pi greco1. Il numero irrazionale pi greco è da sempre uno degli argomenti più affascinanti del mondo dell'analisi matematica.

Per chi non lo sapesse questa entità numerica è data dal rapporto esistente tra la circonferenza e il diametro di un cerchio.

Ora vi proporremo 2 differenti metodi di calcolo di pi greco.

Il matematico inglese John Wallis ha dimostrato che :

(PI GRECO)/2=(2/1)*(2/3)*(4/3)*(4/5)*(6 /5)*(6/7).....

Tramite questo programma, il cui algoritmo è fondato sulla formula precedentemente enunciata, sarà possibile avvicinarsi progressivamente al valore di questo numero misterioso.

(La Redazione)

```
100 PI=2:N=2
110 PI=PI*N/(N-1)*N/(N+1)
120 PRINTN,PI:N=N+2:GOTO 110
```

$ 4E

Pi greco2. Il secondo metodo per il calcolo di pi greco è basato sulla formula enuciata e dimostrata dal matematico John Machin:

(PI GRECO)/4=4*ATN(1/5)-ATN(1/239)

Il simbolo ATN denota la funzione trigonometrica arcotangente.

Attenzione!! Chiunque riesca a trovare una periodicità nella disposizione dei decimali di questo numero, è pregato di comunicarcelo immediatamente.

Naturalmente saremo ben lieti di ospitare sulle pagine della

Se infatti quell'individuo sarà in possesso della medesima routine da noi utilizata, potrà appurare al termine di ogni linea del nostro software se ha effettuato la digitazione in maniera corretta.

```
100 PRINT"[CLEAR]ATTENDERE PREGO..
    .":FOR I=886 TO 1018:READ A:CK
    =CK+A:POKE I,A:NEXT
110 IF CK<>17539 THEN PRINT"[DOWN]
    ERRORE":END
120 SYS886:PRINT"[CLEAR][2 DOWN]AT
    TIVATO":END
130 DATA 173,36,3,201,150,208
140 DATA 1,96,141,151,3,173
150 DATA 37,3,141,152,3,169
160 DATA 150,141,36,3,169,3
170 DATA 141,37,3,169,0,133
180 DATA 254,96,32,87,241,133
190 DATA 251,134,252,132,253,8
200 DATA 201,13,240,17,201,32
210 DATA 240,5,24,101,254,133
220 DATA 254,165,251,166,252,164
230 DATA 253,40,96,169,13,32
240 DATA 210,255,165,214,141,251
250 DATA 3,206,251,3,169,0
260 DATA 133,216,169,19,32,210
270 DATA 255,169,18,32,210,255
280 DATA 169,58,32,210,255,166
290 DATA 254,169,0,133,254,172
300 DATA 151,3,192,87,208,6
310 DATA 32,205,189,76,235,3
320 DATA 32,205,221,169,32,32
330 DATA 210,255,32,210,255,173
340 DATA 251,3,133,214,76,173
350 DATA 3
```

$ 5B

Usa il wait. Esiste un'istruzione nel BASIC che non viene usata con eccessiva frequenza, anzi mai! WAIT viene usato per interrompere il programma fino a quando i contenuti di una locazione di memoria non cambiano in modo specifico.

In questo programma premendo il tasto "fire" del Joystick si ottiene un effetto di lampeggio. Infatti viene mutato il va lore della locazione di memoria che sovraintende all'interpretazione del movimento effettuato con il supporto giochi.

(La Redazione)

```
100 PRINT"[UP]
          "
110 PRINT"[UP]ACQUISTA COMMODORE C
    LUB"
120 WAIT 56464,16
130 GOTO 100
```

$ 5C

Numero e. Tramite questa breve routine vengono calcolati i primi decimali del numero e.

Questa entità matematica costituisce insieme a pi greco e a I uno degli argomenti più interessanti nell'ambito dello studio dei numeri irrazionali.

(La Redazione)

```
100 P=23300
110 E=(1+1/P)↑(P)
120 PRINT"E="E
```

$ 5D

Caratteri. Questa breve routine vi permetterà di visualizzare, in rapida successione, tutti i caratteri grafici disponibili sul Commodore 64.

```
100 REM  G.B. SOFTWARE 1985
110 DATA 169,0,234,234,162,0,157,0
    ,4,157,0,5,157,0,6,157,0,7,232
    ,208,241,200,208,1
120 DATA 96,152,160,0,162,0,232,20
    8,253,200,208,248,168,76,4,16,
    234
130 FOR K=4096 TO 4136:READ A:POKE
     K,A:NEXT
140 SYS4096
```

$ 5E

Dimezza il tempo di lettura. Il Commodore 64 adotta un sistema di registrazione diverso rispetto a quello utilizzato dagli altri calcolatori.

Digitando queste istruzioni potrete dimezzare i tempi di lettura da nastro:

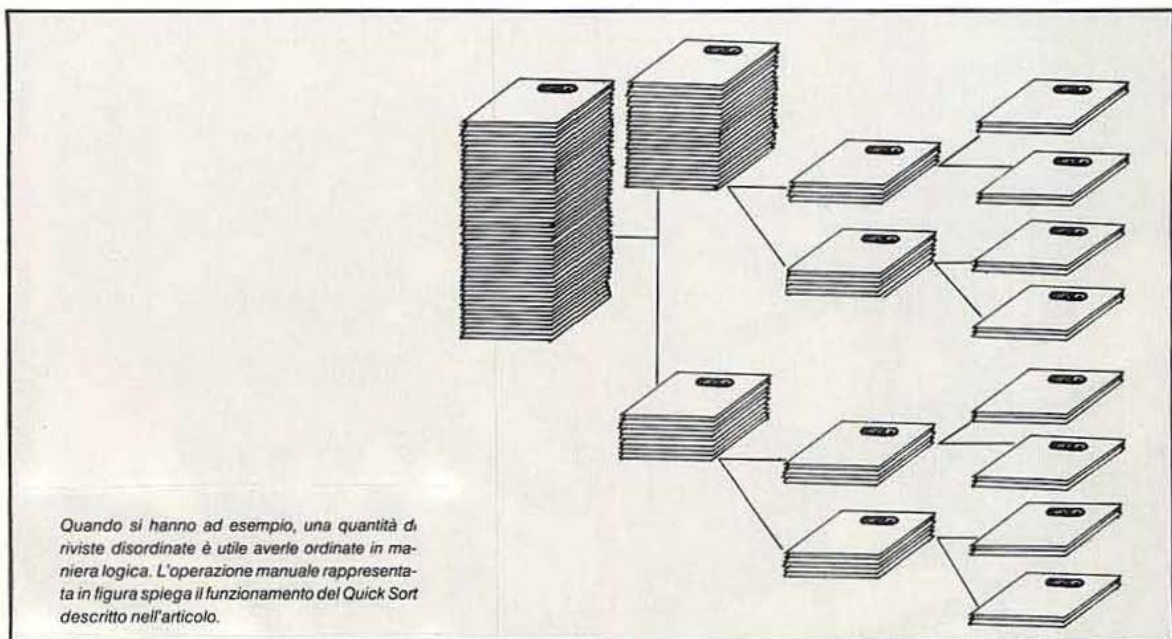
```
POKE43,PEEK(829):POKE44,PEEK(830):POKE45,

PEEK(831):POKE46,PEEK(832):CLR:RESTORE
```

(La Redazione)

SORT

di **Ernesto Sidoti, Rinaldo Nani**

Quando si hanno ad esempio, una quantità di riviste disordinate è utile averle ordinate in maniera logica. L'operazione manuale rappresentata in figura spiega il funzionamento del Quick Sort descritto nell'articolo.

Per qualche strano motivo, tutto ciò che ci circonda "tende" al disordine. Per esempio gli oggetti di una stanza si sparpagliano "spontaneamente" se non ci curiamo di riporli al loro posto. E i pianeti, le idee, i dati su di un file seguono spesso la stessa SORTe. Ancora più strano è che l'uomo e la natura terrestre vadano contro corrente: dal DNA alla struttura cristallina dei fiocchi di neve, tutto sembra seguire un Ordine con la "O" maiuscola, una qualche regola anti-sparpagliamento.

E' possibile lavorare su di una scrivania totalmente in disordine, ma è evidente che non si può pensare costruttivamente con tutte le idee confuse e quindi l'ordine deve necessariamente esistire ad un qualche livello. Forse è per questa necessità di ordine che abbiamo affinato, dentro di noi, sitemi di ordinamento molto complessi e potenti.

Trovandoci davanti uno scaffale di libri messi a caso ci basta una occhiata per farci un'idea di come metterli in ordine alfabetico per autore: cominceremo così a muovere uno o più libri dallo scaffale spostandoli di qua e di là ed inserendoli in posizioni più o meno "ragionevoli" ripeteremo queste operazioni finchè lo scopo non sarà raggiunto.

Non è difficile immaginare un programma che sappia ordinare un insieme di dati secondo una certa CHIAVE: un programma siffatto si chiama SORT (tra parentesi, nell'esempio dei libri sullo scaffale la CHIAVE era il nome dell'autore: si sarebbe potuto benissimo ordinare gli stessi libri secondo l'altezza o secondo l'argomento cambiando così la CHIAVE di ordinamento). E' chiaro che concetti come "dare un'occhiata", "farsi un'idea", sono difficilmente programmabili (se qualcuno ci riesce si faccia vivo !).

Rimane comunque il fatto che i SORT, proprio perchè li ha fatti l'uomo, imitano il nostro modo di agire. Di SORT ne esistono tanti, e in questo articolo viene proposto solo qualche esempio. Poi, una volta capito il concetto, ognuno potrà programmarsi il suo SORT personalizzato.

Ampliando l'esempio dei libri sullo scaffale immaginiamo di dover ordinare una grande biblioteca a più piani. Ogni libro ha il suo titolo, l'autore ed un numero di riferimento.

Esempio:

Dante/La Divina Commedia/78
Apostol/Analisi I /234

....

ecc.

Scegliamo come chiave di ordinamento il numero associato ad ogni libro.

Avendo premesso che la biblioteca è a più piani non possiamo avere una visione globale di tutti libri e a maggior ragione non possiamo stabilire a priori la posizione definitiva di un libro. Ci serve un metodo semplice e sicuro, non importa in questo caso se è lento, per ordinare l'intera biblioteca. Un sistema potrebbe essere quello di prendere in considerazione il numero di riferimento del primo libro della serie e verificare se il numero di riferimento del libro successivo è minore . Se questa situazione è soddisfatta scambiamo la posizione dei due libri. Altrimenti confrontiamo il numero di riferimento del secondo libro con quello del terzo volume. A questo punto nuovamente ci chiediamo se la condizione precedentemente analizzata si verifica e in tal caso operiamo lo scambio. E cosi via per tutti i libri della biblioteca.

La situazione sarà più chiera con un esempio numerico. Supponiamo di avere questa serie di numeri di riferimento:

78 234 8 45 2 1 .... ecc.

Ci chiediamo: "234 è maggiore o uguale a 78 (ciò equivale a chiederci se 78 minore di 234)?". La risposta ovviamente è "si" e allora non operiamo lo scambio. La serie quindi rimane inalterata.

78 234 8 45 2 1 .... ecc.

Ci chiediamo nuovamente: "8 è maggiore o uguale a 234?". La risposta è "no" e questa volta operiamo lo scambio:

78 8 234 45 2 .... ecc.

Ci chiediamo nuovamente: "45 è maggiore o uguale a 234". La risposta è "no", quindi operiamo lo scambio:

78 8 45 234 2 1 .... ecc.

Ci chiediamo nuovamente "2 è maggiore o uguale a 234?". La risposta è "no", quindi operiamo lo scambio:

78 8 45 2 234 1 .... ecc.

avrai già indovinato cosa succederà dopo:

78 8 45 2 1 234 .... ecc.

Operando così per tutti i numeri di riferimento della biblioteca, alla fine avremo nell'ultima posizione il volume con un numero di riferimento più alto e raggiunto un più alto livello di Ordine. Per raggiungere il completo ordinamento si dovrà ripetere il medesimo procedimento tante volte quanti sono i libri da ordinare. L'algoritmo descritto, in informatica è comunemente chiamato BUBBLESORT ed è mostrato nella figura 2. Nella figura 3 ci sono tutti i passaggi eseguiti dal programma per ordinare la serie dei 6 numeri prima citati.

Ci vorranno diversi mesi (che dico? diversi anni) per riordinare con questo metodo la nostra biblioteca. Allo stesso modo una macchina che esegue questo algoritmo impiegherà molto tempo per ordinare una lunga sequenza di numeri. La semplicità dell'algoritmo nasconde (ma non troppo) una notevole inefficienza: ad esempio elaborando una serie già ordinata continuerebbe a confrontare coppie di numeri inutilmente. La soluzione di questo specifico problema è descritta in calce alla figura 2. Per chi è pratico in matematica basta dire che il tempo di esecuzione è O(n elevato a 2) dove nè il numero di dati in ingresso; quindi l'algoritmo appena esaminato si rivela efficente per piccole serie di dati non lo sarà per serie molto lunghe. Gli algoritmi derivati da una ottimizzazione del BUBBLESORT sono molti.

```
10 REM  CARICAMENTO VETTORE DA OR
   DINARE
20 PRINT"[CLEAR]SCRIVI IL NUMERO
   MASSIMO DI ELEMENTI"
30 INPUT  NDATI
40 DIM  MAX(NDATI)
50 FOR  S=1  TO  NDATI
60 PRINT"DATO NUMERO";S;:INPUT  M
   AX(S)
70 NEXT S
80 PRINT"LISTA DA ORDINARE":GOSUB
    90:END
90 PRINT:FOR  S=1  TO  NDATI
100 PRINTMAX(S)
110 NEXT:RETURN
```

Fig. 1: Questo piccolo programma servirà per caricare il vettore da ordinare. Per tutti i Sort usati si suppone di ordinare sempre il vettore MAX e di disporre di un numero di elementi pari a N Dati.

Uno fra i più conosciuti è lo SHAKESORT. Nasce dal fatto che memorizzando il più grande indice al di sotto del quale gli elementi sono in ordine, nella

```
100 REM  BUBBLE1
110 :FOR  A=1  TO  NDATI
120 ::FOR  B=1  TO  NDATI-A
130 ::::IF  MAX(B+1))=MAX(B)  THEN
     170
140 :::::X=MAX(B)
150 :::::MAX(B)=MAX(B+1)
160 :::::MAX(B+1)=X
170 ::NEXT B
180 :NEXT A
190 END


  0 REM  ** OTTIMIZZAZIONE **
125 FLAG=0
165 FLAG=1
175 IF  FLAG=0  THEN  180
```

Fig. 2.1: In questo listato è riportato il classico Bubble Sort a cui possiamo inserire l'ottimizzazione suggerita.

```
A=1
  - INIZIO   78  234    8   45    2    1
      B=1    78  234    8   45    2    1
      B=2    78    8  234   45    2    1
      B=3    78    8   45  234    2    1
      B=4    78    8   45    2  234    1
      B=5    78    8   45    2    1  234

A=2
      B=1     8   78   45    2    1  234
      B=2     8   45   78    2    1  234
      B=3     8   45    2   78    1  234
      B=4     8   45    2    1   78  234

A=3
      B=1     8    2   45    1   78  234
      B=2     8    2    1   45   78  234
      B=3     8    2    1   45   78  234*
A=4
      B=1     2    8    1   45   78  234
      B=2     2    1    8   45   78  234
A=5
      B=1     1    2    8   45   78  234
```

Fig. 3: I numeri citati in inizio vengono trattati da Bubble Sort in questa maniera. Osservate in particolare l'andamento diagonale degli spostamenti (questo ciclo viene saltato se l'algoritmo è ottimizzato).

scansione successiva ci si può fermare in quella posizione. Il programma SHAKESORT è illustrato nella figura 4.

Un altro metodo di ordinamento è chiamato STRAIGHT INSERTION SORT. Vediamo il suo uso con un esempio forse un pò banale: sistemare le carte quando gioco a scala quaranta. Possiamo, ad esempio, ordinare 6 carte formando dei tris o raggrupparle secondo colore ma per semplicità supponiamo di ordinarle secondo ordine ascendente. Se ho in mano queste carte:

10 13(K) 9 8 11(J) 12(Q)

Trovo più naturale prendere il 9 e met-

terlo davanti al 10 di modo da ottenere:

10 13(K) 8 11(J) 12(Q)

Quindi prendo l'8 e lo sposto davanti al 9; poi prendo il fante (11) e trovo la sua posizione tra il 10, e il Re, e così via fino ad ottenere:

8 9 10 11(J) 12(Q) 13(K)

```
100 REM  SHAKESORT
110 L=2:R=NDATI:K=NDATI
120 :FOR  J=R  TO  L  STEP -1
130 ::IF  MAX(J-1)(MAX(J)  THEN  1
    80
140 :::X=MAX(J-1)
150 :::MAX(J-1)=MAX(J)
160 :::MAX(J)=X
170 :::K=J
180 :NEXT J
190 L=K+1
200 :FOR  J=L  TO  R
210 ::IF  MAX(J-1)(MAX(J)  THEN  2
    60
220 :::X=MAX(J-1)
230 :::MAX(J-1)=MA(J)
240 :::MAX(J)=X
250 :::K=J
260 :NEXT J
270 R=K-1
280 :IF  L(=R  THEN  120
290 END
```

FIGURA 4

Formalizziamo il metodo aiutandoci con lo schema 5. Prendiamo in considerazione il secondo numero, e ne cerchiamo l'esatta posizione tra i numeri che stanno alla sua sinistra: in questo caso il secondo numero è 13 e l'unico numero presente alla sua sinistra è il 10, quindi il 13 rimane dov'è.
Passiamo al terzo numero (il 9); considerati i numeri alla sua sinistra, la sua posizione nel vettore dovrà essere la prima. Per ottenere ciò spostiamo il 13 in

```
100 REM   STRAIGHT INSERTION SORT
110 :FOR  I=2  TO  NDATI
120 :::X=MAX(I)
130 :::MAX(0)=X
140 :::J=I-1
150 :::IF  X)=MAX(J)  THEN  190
160 ::::MAX(J+1)=MAX(J)
170 ::::J=J-1
180 :::GOTO  150
190 :::MAX(J+1)=X
200 :NEXT I
210 END
```

Fig. 5: Questa implementazione del programma e eseguibile in BASIC. Come si può notare si accetta lo 0 come primo indice dell'array. Esistono però alcuni interpreti BASIC che non accettano l'indice 0 quindi i valori da ordinare devono essere memorizzati nel vettore MAX a partire dall'indice 2 e bisogna incrementare il valore d'inizio della FOR di 1. E' palese che la riga 40 leggerà MAX (1) = X.

ORDINE INIZIALE	10	13(K)	9	8	11(J)	12(Q)
I=2	10	13(K)	9	8	11(J)	12(Q)
I=3	9	10	13(K)	8	11(J)	12(Q)
I=4	8	9	10	13(K)	11(J)	12(Q)
I=5	8	9	10	11(J)	13(K)	12(Q)
I=6	8	9	10	11(J)	12(Q)	13(K)

Fig. 5.1: Listato della routine del Sort che è una evoluzione del Bubble Sort.

avanti nella precedente posizione del 9, il 10 nella precedente posizione del 13 ed il 9 infine in testa alla serie. Con ragionamento analogo ordiniamo il quarto, il quinto ed il sesto numero. Questo è il procedimento che esegue il programma STRAIGHT INSERTION SORT della figura 5.

Quarto ed ultimo sort "semplice" è lo STRAIGHT SELETION SORT: anche per questo algortmo il funzionamento è intuitivo. Guardiamo la figura 6. Si cerca il numero più piccolo della serie disordinata e lo si scambia con il primo elemento del vettore. Nella prima riga dell' esempio il più piccolo è il 2 che si scambia con l'11. Poi si prende il secondo elemento, il 26 e si cerca il più piccolo numero tra quella alla sua destra: in questo caso il 4. Si effettua lo scambio. Poi si prende il terzo e si cerca il minimo alla sua destra e così via fino ad ottenere la serie ordinata.

Tutti gli algoritmi presentati fino ad ora hanno il difetto di essere inefficenti nell'ordinamento di lunghe serie, non solo perchè la quantità di numeri è elevata ma anche perchè, incremetando di poco il numero degli elementi da ordinare, si aumenta eccessivamente il tempo di esecuzione. Chiaramente lo SHAKESORT ordinerà una serie molto più velocemente di uno STRAIGHT INSERTION SORT (vedi tabella). Ma questi sort, pur comportandosi in modo più o meno efficente, appartengono alla famiglia O(N elevato 2). Per esempio il BUBBLESORT e lo STRAIGHT SELECTION SORT effetuano il confronto sulle coppie di elementi adiacenti. Se i dati sono molto disordinati possono essere necessari al più n-1 passaggi per i dati non ancora ordinati. Questo può richiedere fino a "N elevato 2 fratto 2" confronti (da qui O(N elevato a 2)). Ciò significa che se nt200 solo i confronti sono 20000 e basta aumentare il numero dei dati da sortare di 50 unità che i confronti diventano 31250!

Se vogliamo raggiugere una maggiore efficenza non basta diminuire il numero degli scambi, ma bisogna creare degli algoritmi che all'aumentare dei dati da ordinare non aumentino in modo sproporzinato il numero dei confroti e degli scambi. Un passo avanti in questa direzione si ha con l'algortmo SHELLSORT. Il principio di funzionamento è di confrontare elementi molto distanti fra loro nella speranza che uno scambio effettuato su tali elementi, migliori l'efficienza dell'algoritmo rispetto ad un altro che effettua scambi su elementi contigui.

La distanza tra due elementi da confrontare si chiama "gap". Questa distanza viene dimezzata ad ogni ciclo quando il gap raggiungerà il valore 1 il programma si comporterà come il BUBBLESORT. Ma questa volta i dati sono quasi in ordine e la passata sarà velocissima. Il programma è riportato nella figura 7 e per aver chiaro il funzionamento consiglimo di eseguire manualmente l'algoritmo. Lo SHELLSORT è poco indicato per ordinare piccole serie di numeri (impiega spesso un tempo maggiore che un BUBBLESORT); la sua efficienza si nota su lunghe serie in quanto il suo comportamento non è più un O(n elevato 2), ma bensì un O(n elevato 1.5) (vedi figura).

Un algoritmo ancor più intelligente è il QUICKSORT. Per spiegarlo mi riferirò ad una banale esperienza personale. Qualche tempo fa cominciai a raccoglie-

```
100 REM  STRAIGHT SELECTION SORT
110 :FOR  I=1  TO  NDATI-1
120 :::K=I:X=MAX(I)
130 :::::FOR  J=I+1  TO  NDATI
140 :::::::IF  MAX(J)>X  THEN  170
150 ::::::::K=J
160 ::::::::X=MAX(J)
170 :::::NEXT J
180 :::MAX(K)=MAX(I):MAX(I)=X
190 :NEXT I
200 END
```

ORDINE INIZIALE	11	26	4	2	16	9
I = 1	11	16	4	2	16	9
I = 2	2	26	4	11	16	9
I = 3	2	4	26	11	16	9
I = 4	2	4	9	11	16	26
I = 5	2	4	9	11	16	26

Fig. 6: Ecco l'ultimo Sort della serie semplice (O(n). Osservate come avviene la ricerca dell'elemento insieme al quale, una volta trovato, viene scambiato con quello presente nella scansione.

re i fascicoli settimanali di una grande opera, una delle tante vendute in edicola. Era mia abitudine, prima di andare a letto, di leggere il fascicolo appena acquistato. Puntualmnte detto fascicolo seguiva la "legge del disordine". Dopo molti mesi arrivò l'ultimo numero. Decisi che valeva la pena di rilegareli tutti e, dopo un'affannosa ricerca per tutta la casa mi ritrovai solo con un'interminabile pila di fascicoli da ordinare.

Ho preso un fascicolo a caso dalla pila e ho accatastato da una parte quelli con un numero inferiore ad esso e dall'altra quelli con un numero maggiore. Quindi ho diviso le due cataste cosi' ottenute in modo analogo ottenendone 4. Continuando ad applicare questo sistema sono arrivato all'ordine desiderato.

A prima vista sembrerebbe facile formalizzare un comportamento simile; purtroppo i controlli spontanei che ho applicato nell'ordinare i fascicoli sono tutt'altro che semplici. E sono altrettanto complessi i passi eseguiti da un calcolatore per ordinare con un criterio simile una lista di numeri.

Ecco come potrebbe essere un vettore di 6 elementi da ordinare:

```
100 REM  ** SHELL SORT **
110 FLAG=0
120 V=0:T=0
130 X=0:K=NDATI
140 IF  K<=1  THEN  290
150 ::::K=INT(K/2)
160 ::::T=NDATI-K
170 ::::FLAG=0
180 :::::FOR  J=1  TO  T
190 :::::::V=J+K
200 :::::::IF  MAX(J)<=MAX(V)  THE
    N  250
210 :::::::X=MAX(J)
220 :::::::MAX(J)=MAX(V)
230 :::::::MAX(V)=X
240 :::::::FLAG=1
250 :::::NEXT J
260 :::IF  FLAG=0  THEN  280
270 :GOTO  170
280 GOTO  140
290 END
```

Fig. 7: Il Sort tipo Shell può essere considerata più diffusa routine di ordinamento di tipo complesso.

MAX(1)	MAX(2)	MAX(3)
25	32	77

MAX(4)	MAX(5)	MAX(6)
90	8	54

Ed ecco come opera il QUICK

- Definire i puntatori A e B alle due estremità del vettore; MAX(A) diventa il "pivot" (nell'esempio dei fascicoli il pivot era il numero preso a caso).

- Si confronta il pivot con il valore del puntatore opposto (nel nostro esempio il 25 viene confrontato con 54).

- Scambiare i due numeri se non sono in giusto ordine (nell'esempio il 25 e il 54 NON si scambiano).

- Spostare il puntatore opposto al pivot di un posto verso di esso (nell'esempio B=B-1).

- Se A=B vai a 6 se no torna a 2. Infatti se AtB tutti i numeri sono stati confrontati e si sono ottenute due liste separate dal pivot che si trova nella sua posizione definitiva.

- Ripeti da 1 a 5 per ciascuna sottolista (tornando al passo 1 si definirà un altro pivot)

Con qualche trucco si possono tradurre facilmente in un FLOW-CHART i concetti contenuti nei passi da 1 a 5 (vedi fig. 8). Ad esempio il concetto espresso dai passi 2 e 4: "il puntatore opposto al pivot" è stato tradotto con l'ausilio della variabile S che assume il valore 1 se il pivot è nella parte B della lista e il valore -1 se è nella parte A. Ogni volta che il pivot viene scambiato di posto S diventa -S. Un altro semplice artificio sono le variabili A1 e B1 che rappresentano gli estremi della lista o sottolista considerata. I guai cominciano quando si cerca di formalizzare il passo 6; il vero problema è far sì che il computer si "ricordi" di tutte le sottoliste che ha creato, per poi poterle analizzare una alla volta. Mi spiego meglio: supponiamo che il programma abbia diviso (operando come descritto dal FLOW-CHART della figura 8) la lista dell'esempio ottenendo:

Sottolista 1		PIVOT	Sottolista 2		
8	9	25	77	32	54
A1	A-1	A	A+1		B1

Gli estremi delle 2 nuove sottoliste sono A1,A-1 e A+1,B1. Il QUICK-SORT lavorando su una sola sottolista alla volta, deve memorizzare i confini de'la sottolista rimanente. E se divide ancora la sottolista in diverse sotto-sottoliste, dovrà memorizzarne da qualche parte tutti i confini. Dovremo quindi costruirci uno STACK. Lo STACK è una pila di dati del tipo "last in first out" (l'ultimo dato che entra è il primo a uscire); un modo per visualizzare uno STACK è quello di pensare ad una pila di piatti lavati, ma ancora da asciugare. L'ultimo piatto lavato si trova in cima alla pila e sarà il primo ad essere asciugato.

Lo STACK che utilizziamo nel programma è un array così definito:

DIM STACK(10,2)

ed è uno STACK a 2 colonne (una doppia pila di piatti). Ogni volta che si crea una nuova sottolista il puntatore PNT aumenta di 1 e STACK(PNT,1) as-

Fig. 8: Rappresentazione in diagramma di flusso dalla subroutine dal Q Sort di cui possiamo osservare l'utilizzo corretto nella figura 9.

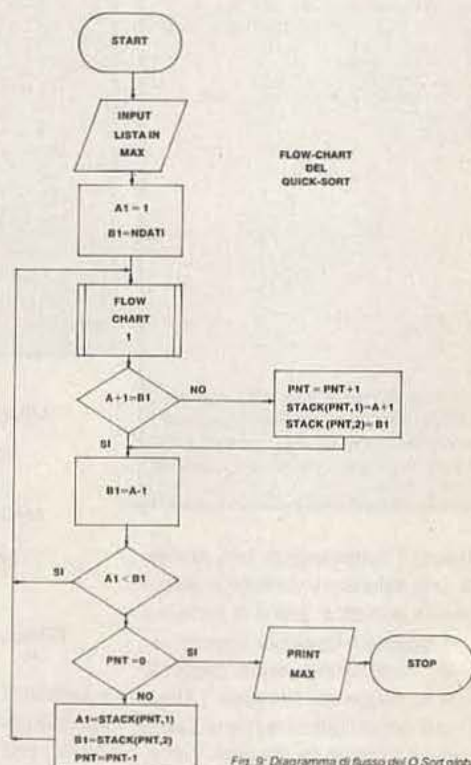
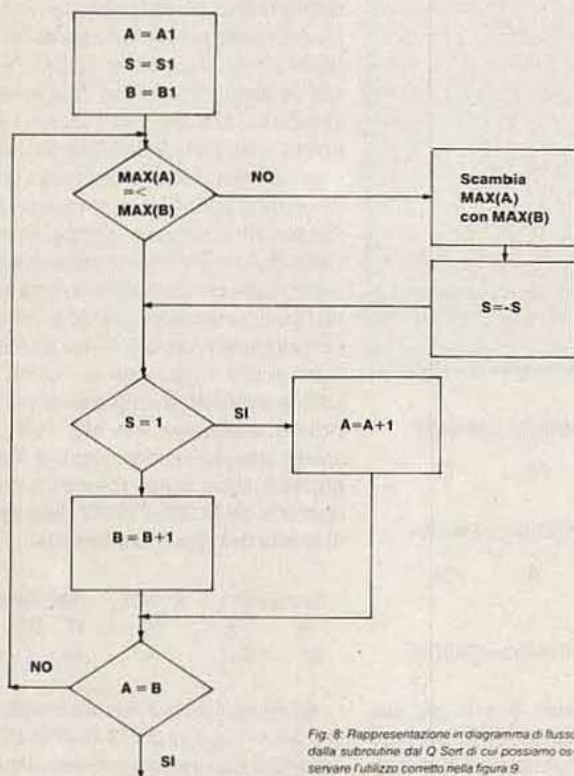
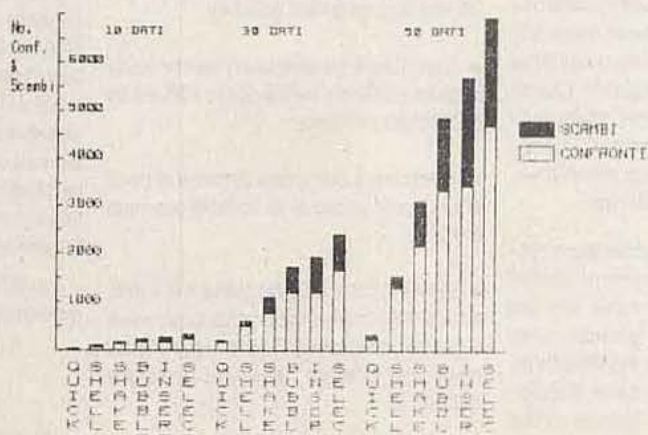
Fig. 9: Diagramma di flusso del Q Sort globale.

```
1000 REM ****************************
1010 REM *                          *
1020 REM *        QUICKSORT         *
1030 REM *                          *
1040 REM ****************************
1050 PNT=1
1060 STACK(PNT,1)=1
1070 STACK(PNT,2)=NDAT1
1080 IF  PNT<=0  THEN  1140
1090 A1=STACK(PNT,1)
1100 B1=STACK(PNT,2)
1110 PNT=PNT-1
1120 GOSUB  1150
1130 GOTO   1080
1140 END
1150 T=0
1160 IF  B1<=A1  THEN  1370
1170 A=A1
1180 B=B1
1190 S=-1
1200 IF  A>=B  THEN  1310
1210 IF  MAX(A)<=MAX(B)  THEN  1260
1220 T=MAX(A)
1230 MAX(A)=MAX(B)
1240 MAX(B)=T
1250 S=-S
1260 IF  S<0  THEN  1290
1270 B=B-1
1280 GOTO  1300
1290 A=A+1
1300 GOTO  1200
1310 IF  A+1)=B1  THEN  1350
1320 PNT=PNT+1
1330 STACK(PNT,1)=A+1
1340 STACK(PNT,2)=B1
1350 B1=A-1
1360 GOTO  1160
1370 RETURN
```

Fig. 10: Il listato rappresenta la subroutine Q Sort ricambiabile tramite GOSUB. Osservate il programma che è l'esatta traduzione in BASIC del Flow Chart di figura 9 e 8.

sume il valore del confine sinistro e STACK(PNT,2) assume il valore del confine destro. Quando il programma ha terminato il lavoro su una sottolista, andrà a prendere dallo STACK i confini della sottolista da analizzare successivamente. Una volta analizzate tutte le sottoliste indicate dallo STACK i numeri saranno in ordine. L'intero procedimento è illustrato dal FLOW-CHART della figura 9. Nella figura 10 è riportato il programma QUICKSORT in BASIC.

Non bisogna da scoraggiarsi se questo algoritmo sembra astruso: il modo per aver ben chiaro il suo funzionamento è ancora una volta quello di eseguirlo con carta e matita. I nostri sforzi però non sono stati vani: dando un'altra occhiata alla tabella 1 ci si accorge che il QUICKSORT non ha rivali nell'ordinare lunghe serie di numeri. Matematicamente il nostro entusiamo si riassume così: il QUICKSORT è in molti casi O(n)!.

Fermiamoci un attimo per riprendere fiato ed esaminiamo quello che abbiamo fatto. Con l'ausilio di programmi siamo arrivati all'ordine ma con la o minuscola, anche se un programma come il QUICKSORT è difficilmente battibile come velocità e relativa semplicità. Il fatto è che fino ad ora abbiamo ordinato serie di numeri effettuando confronti e scambi. Spesso invece i dati da ordinare sono RECORDS (Il record è un insieme di elementi che riguardano il dato. ESEMPIO: in una rubrica telefonica ogni record è composto da **3** campi: NOME/INDIRIZZO/TEL.) e l'ordinamento viene eseguito su di un campo chiave. Il confronto di due campi alfanumerici è equivalente al confronto di 2 numeri.

Sembrerebbe opportuno usare uno dei SORT descritti sopra per effettuare un ordinamento di records. Ma quando scambiamo due records sorgono dei problemi: dovremmo spostare tutti i campi aumentando di molto i tempi di esecuzione. Quello che serve in questi casi è un SORT che non effettua alcun scambio. Un esempio:

nome(*)	bellezza	disponibilità	n. tel.	xx
Maria	7.5	9.5	439080	
Sonia	5	6	404425	
Elena	3	4	204356	
Donata	7	7	469168	
Stefy	9	6.5	468196	
Anton	8	0	230577	
Dora	10	-1	378995	
....	.....	.....	....	

Supponiamo che voglia ordinarla per ordine di bellezza. Sarebbe comodo che in qualche modo si riempisse la colonna xx fino ad ottenere la tabella:

nome(*)	bellezza	disponibilità	n. tel.	xx
Maria	7.5	9.5	439080	4
Sonia	5	6	404425	6
Elena	3	4	204356	7
Donata	7	7	469168	5
Stefy	9	6.5	468196	2
Anton	8	0	230577	3
Dora	10	-1	378925	1
....	.....	.....	....	...

(*) Nomi e tel. sono casuali!

Se stampiamo i nomi riferendoci all'ordine xx li avremo ordinati secondo bellezza (vedi fig. 11 per l'algortmo).

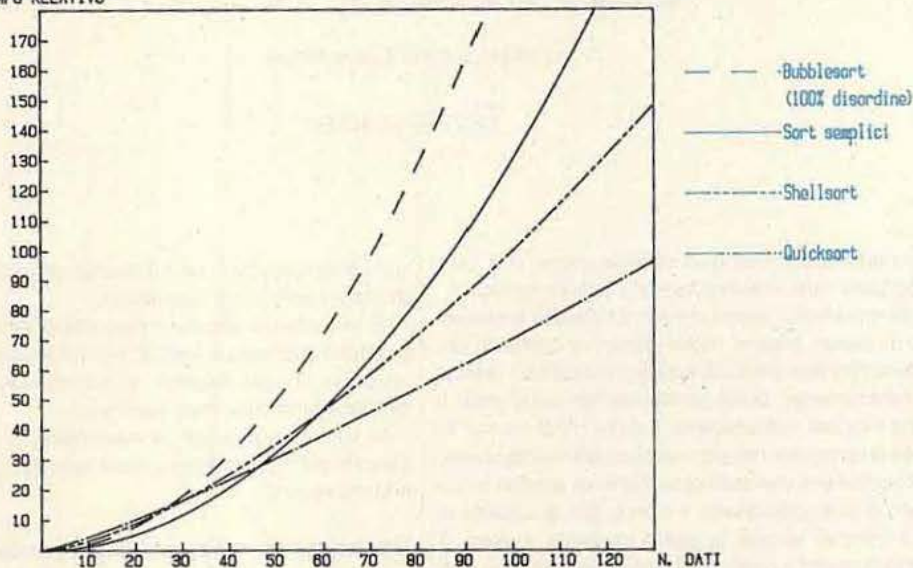
Ultimo consiglio: invece di accumulare dati in modo disordinato (per poi doverli ordinare), create una procedura che ordini immediatamente il dato appena enterato.Questo trucco è contro ogni disordine di SORTa. Per essere sicuri di scegliere il SORT adatto per il vostro software vi basta digitare il programma in coda all'articolo. Questo fornisce la possibilità di generare dei numeri caso e di definire la percentuale d'ordine dei dati (0 totalmente disodinati, 100 tutti già ordinati) e infine eseguirà 6 tipi di ordinamento e stamperà per ognuno il numero dei confronti, il numero degli scambi ed il tempo impiegato per raggiungere l'ordine.

```
  5 DIM  MAX$(50):DIM  G(50)
 10 PRINT"SCRIVI IL NUMERO DEI DAT
    I";
 20 INPUT  NDATI
 30 FOR A=1  TO  NDATI
 40 INPUT  MAX$(A)
 50 NEXTA
 60 GOSUB  100
 70 FOR  A=1  TO  NDATI
 80 PRINTMAX$(G(A))
 90 NEXT A
 95 END
 96 ***************************
 98 *  OR DINAMEN TO  PER INDICE *
 99 ***************************
100 FOR  A=1  TO  NDATI
110 PUNT=1
120 FOR  B=1  TO  NDATI
130 IF  MAX$(A)>MAX$(B)  THEN  PUN
    T=PUNT+1
150 IF  MAX$(A)=MAX$(B)  AND  A>B
     THEN  PUNT=PUNT+1
160 NEXT B
170 G(PUNT)=A
180 NEXT A
200 RETURN
```

Fig. 11: L'ordinamento per indice può essere considerato un sistema di Sort che più si avvicina al concetto più evoluto di gestione dati su disco. I record non vengono spostati (notevole risparmio di tempo), ma viene creata una tabella ordinata di puntation secondo i pesi dei records stessi.

```
1000 REM ***************************
1020 REM *   S O R T - M A C H     *
1040 REM *           DI            *
1050 REM * ERNESTO       RINALDO   *
1060 REM *            &            *
1070 REM *   SIDOTI        NANI    *
1090 REM ***************************
1100 :
1110 PRINT"[CLEAR][UP]":POKE  53280
     ,0:POKE 53281,0
1120 PRINT"[BIANCO][UP]":U=6
1130 PRINT TAB(U)"[UP] ──────────
     ──────────"
1140 PRINT TAB(U)"──────── SORT-MAC
     H ────────"
1150 PRINT TAB(U)"  ────────────
     ─────────"
1160 PRINT
1170 PRINT"NUMERO DATI           "
1180 PRINT
1190 PRINT"PERCENTALE DI ORDINE
     %"
1200 PRINT
1210 PRINT"┌────────┬──────────────┬
     ──────────┬──────────┐"
1220 PRINT"[UP]| SORT   |  CONFRONTI
     | SCAMBI | TEMPO |"
1230 PRINT"[UP]├────────┼──────────
     ──┼──────────┼──────────┤"
1240 FOR A=1  TO  5
1250 PRINT"[UP]|        |
     |        |       |"
1260 PRINT"[UP]├────────┼──────────
     ──┼──────────┼──────────┤"
1270 NEXT
1280 PRINT"[UP]|        |
     |        |       |"
1290 PRINT"[UP]└────────┴──────────
     ──┴──────────┴──────────┘":PRINT"[H
     OME]"
1300 PRINT"[9 DOWN]"
1310 PRINT"[UP]|QUIK"
1320 PRINT"[DOWN]|SHELL"
1330 PRINT"[DOWN]|SHAKE"
1340 PRINT"[DOWN]|BUBBLE"
1350 PRINT"[DOWN]|INSERT"
1360 PRINT"[DOWN]|SELECT"
1370 :PRINT"[HOME]":GOTO  1410
1380 PRINT"[HOME]":FOR  A=1  TO  R-
     1 :PRINT:NEXT
1390 FOR  A=1  TO  C-1:PRINT"[RIGHT
     ]";:NEXT
1400 RETURN
1410 R=4:C=13:GOSUB 1380:PRINT"
                               "
1420 R=4:C=13:GOSUB 1380: INPUT  NDA
     TI
1430 IF  NDATI>999999  OR  NDATI<=0
      THEN  1410
1440 R=5:C=22:GOSUB  1380:PRINT"[RV
     OFF]>":FOR A=1 TO 10:NEXT
1450 GET  A$:IF A$="" THEN  1500
1460 IF  A$=CHR$(13)  THEN  1510
1470 IF  A$<"0"  OR  A$>"9"  THEN
      1500
1480 V=V+1:D$=D$+A$:C=23:GOSUB  138
     0:PRINTD$
1490 IF  V=3  THEN  1510
1500 R=5:C=22:GOSUB  1380:PRINT"[RV
     S]>":FOR A=1 TO 10:NEXT:GOTO
     1440
```

```
1510 PA=VAL(D$)
1520 IF  PA<0  OR  PA>100  THEN  PA
     =0:V=0:D$="":GOSUB 1380:PRINT"
       ":GOTO  1440
1530 DIM  MAX(NDATI+1):DIM  BX(NDAT
     I)
1540 REM **************************
1542 REM *  GENERAZIONE  RANDOM   *
1544 REM **************************
1545 PA=(PA/100)*NDATI
1550 FOR  Q=1  TO  NDATI
1560 IF (RND(1)*NDATI<PA) THEN  BX(
     Q)=Q:GOTO 1580
1570 BX(Q)=INT(NDATI*RND(1))
1580 NEXT
1590 GOSUB 1670:TEMPO=TI:GOSUB 2630
     :GOSUB 3090:GOSUB 3000
1600 GOSUB 1670:TEMPO=TI:GOSUB 2410
     :GOSUB 3090:GOSUB 3000
1610 GOSUB 1670:TEMPO=TI:GOSUB 1890
     :GOSUB 3090:GOSUB 3000
1620 GOSUB 1670:TEMPO=TI:GOSUB 1720
     :GOSUB 3090:GOSUB 3000
1630 GOSUB 1670:TEMPO=TI:GOSUB 2140
     :GOSUB 3090:GOSUB 3000
1640 GOSUB 1670:TEMPO=TI:GOSUB 2280
     :GOSUB 3090:GOSUB 3000
1650 R=15:C=14:GOSUB 1380:PRINT"[UP
     ]**** FINE ****[2 UP]":PRINT"[
     HOME]":FOR ZZ=1 TO 5000:NEXT:
     RUN
1660 REM **************************
1662 REM *     COPIA VETTORE      *
1664 REM **************************
1670 FOR LO=1 TO NDATI:MAX(LO)=BX(L
     O):NEXT
1680 RETURN
1690 REM :FORT=1 TO NDATI:PRINTBX(T
     ):NEXT:END
1700 :
1710 REM **************************
1712 REM * B U B B L E - S O R T  *
1714 REM **************************
1720 FOR  A=1  TO  NDATI
1730 FOR  B=1  TO  NDA I-A
1740 FLAG=0
1750 CK(1)=CK(1)+1
1760 IF  MAX(B+1)>=MAX(B)  THEN  18
     20
1770 CK(2)=CK(2)+1
1780 X=MAX(B)
1790 MAX(B)=MAX(B+1)
1800 MAX(B+1)=X
1810 FLAG=1
1820 NEXT B
1830 IF  FLAG=0  THEN  1840
1840 NEXT A
1850 RETURN
1860 :
1870 :
1880 REM **************************
1883 REM *   S H A K E S O R T    *
1884 REM **************************
1890 L=2:R=NDATI:K=NDATI
1900 :FOR  J=R  TO  L  STEP -1
1910 CK(1)=CK(1)+1
1920 IF  MAX(J-1)<MAX(J)  THEN  198
     0
1930 CK(2)=CK(2)+1
1940 X=MAX(J-1)
1950 MAX(J-1)=MAX(J)
1960 MAX(J)=X
1970 K=J
1980 NEXT J
1990 L=K+1
2000 FOR  J=L  TO  R
2010 CK(1)=CK(1)+1
2020 IF  MAX(J-1)<MAX(J)  THEN  208
     0
2030 CK(2)=CK(2)+1
2040 X=MAX(J-1)
2050 MAX(J-1)=MA(J)
2060 MAX(J)=X
2070 K=J
2080 NEXT J
2090 R=K-1
2100 IF  L<=R  THEN  1900
2120 RETURN
2121 :
2122 :
2130 REM **************************
2132 REM *  STRAIGHT  INSERTION   *
2134 REM **************************
2140 FOR  I=2  TO  NDATI
2150 X=MAX(I)
2160 MAX(0)=X
2170 J=I-1
2180 CK(1)=CK(1)+1
2190 IF  X>=MAX(J)  THEN  2240
2200 CK(2)=CK(2)+1
2210 MAX(J+1)=MAX(J)
2220 J=J-1
2230 GOTO  2190
```

```
2240 MAX(J+1)=X
2250 NEXT I
2260 RETURN
2261 :
2262 :
2270 REM ***************************
2272 REM *  STRAIGHT  SELECTION    *
2274 REM ***************************
2280 FOR  I=1  TO  NDATI-1
2290 K=I:X=MAX(I)
2300 FOR  J=I+1  TO  NDATI
2310 CK(1)=CK(1)+1
2320 IF  MAX(J)>X  THEN  2350
2340 X=MAX(J)
2350 NEXT J
2360 MAX(K)=MAX(I):MAX(I)=X
2370 CK(2)=CK(2)+1
2380 NEXT I
2390 RETURN
2391 :
2392 :
2400 REM ***************************
2402 REM *   S H E L L S O R T     *
2404 REM ***************************
2410 FLAG=0
2420 V=0:T=0
2430 X=0:K=NDATI
2440 IF  K<=1  THEN  2610
2450 K=INT(K/2)
2470 FLAG=0
2480 FOR  J=1  TO  T
2490 V=J+K
2500 CK(1)=CK(1)+1
2510 IF  MAX(J)<=MAX(V)  THEN  2570
2520 CK(2)=CK(2)+1
2530 X=MAX(J)
2540 MAX(J)=MAX(V)
2550 MAX(V)=X
2560 FLAG=1
2570 NEXT J
2580 IF  FLAG=0  THEN  2600
2590 GOTO  2470
2600 GOTO  2440
2610 RETURN
2611 :
2612 :
2620 REM ***************************
2622 REM *   Q U I K S O R T       *
2624 REM ***************************
2630 PNT=1
2640 STACK(PNT,1)=1
2650 STACK(PNT,2)=NDATI
2660 IF  PNT<=0  THEN  2720
2670 A1=STACK(PNT,1)
2680 B1=STACK(PNT,2)
2690 PNT=PNT-1
2700 GOSUB  2750
2710 GOTO  2660
2720 CK(1)=C1
2730 CK(2)=C2
2740 RETURN
2750 T=0
2760 IF  B1<=A1  THEN  2990
2770 A=A1
2780 B=B1
2790 S=-1
2800 IF  A>=B  THEN  2930
2810 C1=C1+1
2820 IF  MAX(A)<=MAX(B)  THEN  2880
2830 C2=C2+1
2840 T=MAX(A)
2850 MAX(A)=MAX(B)
2860 MAX(B)=T
2870 S=-S
2880 IF  S<0  THEN  2910
2890 B=B-1
2900 GOTO  2920
2910 A=A+1
2920 GOTO  2800
2930 IF  A+1>=B1  THEN  2970
2940 PNT=PNT+1
2950 STACK(PNT,1)=A+1
2960 STACK(PNT,2)=B1
2970 B1=A-1
2980 GOTO  2760
2990 RETURN
2991 :
2992 :
3000 FOR DM=1  TO  3
3010 READ  CC
3020 R=RR+9:C=CC:GOSUB 1380:PRINT"[
     UP]"CK(DM)
3030 NEXT
3040 RESTORE
3050 RR=RR+1
3060 RETURN
3070 DATA  10,24,33
3080 ***  CONTA TEMPO ***
3090 XN=(TI-TEMPO)/50
3100 VV$=STR$(XN)
3110 CK(3)=VAL(LEFT$(VV$,7))
3120 RETURN
```

N DATI		QUICK	SHELL	SHAKE	BUBBLE	INSERTION	SELECTION
10	Confronti	25	84	128	176	185	228
	Scambi	10	22	46	70	94	103
	Time	2.10	2.91	2.44	2.58	1.68	1.71
30	Confronti	164	501	790	1226	1244	1685
	Scambi	40	100	305	528	729	743
	Time	10.52	13.60	17.52	22.24	11.62	12.52
50	Confronti	252	1301	2163	3349	3462	4681
	Scambi	101	245	955	1510	2240	2240
	Time	16.90	41.88	55.30	62.32	34.08	33.28
80	Confronti	512	2666	4844	7925	7984	11144
	Scambi	171	475	1979	3563	4967	5185
	Time	33.26	78.53	123.90	160.00	77.00	82.73
100	Confronti	582	3952	7394	12012	12488	15548
	Scambi	248	744	3301	5800	8343	8442
	Time	44.00	114.70	210.40	257.20	128.02	127.62
120	Confronti	734	3769	7647	14388	14907	22041
	Scambi	307	780	4369	7821	11364	11529
	Time	48.92	114.74	353.40	353.40	181.50	172.18

STATISTICA

di **Mariangela Guardione**

Terza parte

In questa puntata prendiamo in considerazione una parte molto importante della statistica: la teoria delle probabilità.

Nella vita quotidiana, anche chi non s'interessa professionalmente di questa branca, molto spesso si domanda che cosa vogliano dire in realtà termini quali "probabilità", "probabile" e "probabilmente" quasi certamente non si è in grado di trovare una risposta soddisfacente. Eppure chi di noi non ha avuto modo di cimentarsi nel gioco della roulette o in pronostici come il totocalcio e di chiedersi cosa s'intenda quando si dice che il rosso è probabile quanto il nero o che la squadra A, esempio il Verona, vincerà la partita contro la squadra B, esempio l'Udinese? La risposta più immediata nel caso della roulette è che il rosso è probabile quanto il nero, in quanto vi sono 18 numeri rossi e 18 numeri neri. Nel caso del totocalcio, invece probabilità significa che la squadra A vincerà "probabilmente" la partita in quanto, ad esempio, la squadra B sta attraversando un periodo di scarsa forma, dato che da sei domeniche non riesce a vincere.

Quindi da questi due esempi si può osservare come vi siano due esplicazioni del concetto di probabilità, diametralmente opposte in quanto nel secondo caso si fa uso di considerazioni di cui il tifoso dispone.

Il primo la definizione classica di probabilità fu Laplace, che diede anche una sistemazione rigorosa ai risultati ottenuti dagli scienziati del XVIII secolo, definendo la probabilità di un evento come il rapporto fra il numero di casi favorevoli a verificarsi dell'evento stesso e il numero dei casi possibili, con la condizione che questi ultimi siano tutti ugualmente possibili. La teoria delle probabilità ebbe notevoli sviluppi sia nel campo della fisica, ed in modo particolare nella teoria cinetica dei gas che in quello della genetica per lo studio dei numerosi fenomeni sociali e biologici.

La teoria delle probabilità, inoltre, insieme alla statistica, è stata intrinsecamente collegata non solo alla matematica pura ma anche all'uso sempre più diffuso dei calcolatori elettronici. Infatti agli studiosi di statistica che sentivano l'esigenza di dare un fondamento matematico alla teoria delle probabilità la matematica moderna offriva gli strumenti adatti quali l'uso della nozione di funzione misurabile e della moderna teoria dell'integrazione.

L'analisi classica, come tutti ben sanno, si era sempre occupata di funzioni continue a differenza dei problemi di probabilità che si occupano di casi discreti.

E' in questo contesto che presenteremo un programma che permette di effettuare i calcoli legati all'analisi combinatoria da un punto di vista didattico, in quanto le videate sono state strutturate per uno scopo esplicativo.

Si vuole ora passare ad esaminare più in dettaglio tutti i concetti più importanti della teoria delle probabilità e dell'analisi combinatoria.

Definizione matematica di probabilità

Si definisce probabilità di un evento E (detto successo), e viene indicata con P(E), il rapporto tra il numero dei casi favorevoli (h) ad E ed il numero dei casi possibili (n), purchè questi ultimi siano tutti ugualmente possibili:

$p=Pr(E)=h/n$

La probabilità che non si verifichi l'evento E, detto insuccesso, viene indicata con la seguente formula:

$q=Pr(nonE)=(n-h)/n=1-h/n=1-p=1-Pr(E)$

In questo modo si ha: $p+q+1$, cioè: $Pr(E)+Pr(nonE)=1$

Bisogna osservare, a questo punto, che la probabilità di un evento è un numero compreso fra 0 e 1. Se un evento non può presentarsi, la sua probabilità è 0; nel caso in cui è certo, la sua probabilità è 1.

Esempio: sia E l'evento nel quale si presentino con un solo lancio di dado i numeri 3 o 4; ci sono sei modi in cui può cadere, in quanto si possono presentare i numeri 1,2,3,4,5 o 6 che sono quindi tutti equiprobabili. Poichè E può presentarsi in due di questi si ha:

$p=Pr(E)=2/6=1/3$

Di conseguenza la probabilità di un insuccesso, cioè di non ottenere in un lancio i numeri 3 o 4, è data da:

$q=Pr(nonE)=1-1/3=2/3$

Va però precisato che la probabilità non si calcola sempre nel modo esposto precedentemente, cioè come rapporto di casi favorevoli su casi possibili, in quanto nel caso di eventi complicati risulta più utile servirsi un'altra definizione statistica di probabilità detta "stimata o empirica", in quanto è data dalla frequenza relativa del presentarsi di un evento quando il numero delle osservazioni è molto grande.

Quindi, in questo caso, la probabilità è il limite della frequenza relativa all'aumentare indefinitamente delle osservazioni. Facciamo un esempio pratico: qual è la probabilità di estrarre da un mazzo di 40 carte una regina?

n=40, f=4 P(E)=4/40=0.1

Oppure: qual è la probabilità di estrarre una pallina nera da una scatola contente 10 palline bianche, 8 nere e due rosse per un totale di 20 palline?

n=20, f=8 P(E)=8/20=0.4

La definizione statistica citata precedentemente è molto utile come strumento pratico di calcolo, ma suscita difficoltà da un punto di vista matematico in quanto in realtà è impossibile che esista un numero che possa essere preso come limite. E' questo il motivo per cui la moderna teoria delle probabilità è stata sviluppata assiomaticamente, nel senso che la sua definizione è un concetto indefinito, come lo sono il punto e la retta nella geometria euclidea.

Probabilità condizionata: eventi indipendenti e dipendenti

Se E1 e E2 sono due eventi, si definisce come "probabilità condizionata" Pr(E2]E1) di E2, posto che E1 si sia presentato, la probabilità che, presentatosi E1, si presenti E2. Se il presentarsi o il non presentarsi di E1 non influisce sulla probabilità del presentarsi dell'evento E2, allora si ha: Pr(E2]E1)=Pr(E2) e si dice che E1 e E2 sono eventi indipendenti, in caso contrario si parla di eventi dipendenti.

Se si denota con E1 e E2 un'evento composto allora si ha:

Pr(E1E2)=Pr(E1)Pr(E2E1) in modo particolare si ha:

Pr(E1E2)=Pr(E1)Pr(E2) nel caso di eventi indipendenti.

Nel caso di tre eventi E1,E2 e E3 si ha:

Pr(E1E2E3)=Pr(E1)Pr(E2E1)Pr(E3E1E2)

Questo rappresenta la probabilità del presentarsi di E1, E2 e E3, che risulta essere uguale alla probabilità di E1 moltiplicato per la probabilità di E2, posto che E1 si sia presentato, moltiplicato ancora per la probabilità di E3 con la condiziowe che gli eventi E1 e E2 si siano presentati. In particolare nel caso di eventi indipendenti si ha:

Pr(E1E2E3)=Pr(E1)Pr(E2)Pr(E3)

Eventi che si escludono a vicenda

Si dice che due o più eventi si escludono a vicenda se il presentarsi di uno di essi esclude il presentarsi degli altri. Tutto questo si esprime con seguente simbologia:

Pr(E1E2)=0

Invece se E1+E2 indica un evento composto si ha:
Pr(E1+E2)=Pr(E1)+Pr(E2)-Pr(E1E2)

e nel caso di eventi che si escludono a vicenda si ha:

Pr(E1+E2)=Pr(E1)+Pr(E2)

Per meglio spiegare si riport.. seguente esempio: se E1 rappresenta l'estrazione di un a.. da un mazzo di carte e E2 è l'evento dell'estrazione di un re, ..lora si ha:

Pr(E1)=4/52=1/13 e
Pr(E2)=4/52=1/13.

Quindi si ottiene che la probabilità di ottenere o un asso o un re in una sola estrazione è data da:

Pr(E1+E2)=Pr(E1)+Pr(E2)=1/13+1/13=2/13

in quanto sia l'asso che il re sono eventi mutuamente esclusivi in quanto non possono essere estratti contemporaneamente.

Precedentemente si è parlato della probabilità di eventi complessi la cui enumerazione può spesso risultare difficile e per questo motivo si utilizza l'Analisi Combinatoria. In particolare il calcolo combinatorio permette di determinare il numero di certi raggruppamenti che si possono ottenere con n oggetti a1,a2,...., anche si supporranno tutti diversi fra loro. I raggruppamenti che verranno considerati sono: le disposizioni, le permutazioni e le combinazioni; essi differiscono fra loro per "la presenza" di almeno un oggetto diverso oppure per il "diverso ordine" con cui gli oggetti si dispongono.

Disposizioni

Il numero A(n,k) delle disposizioni di n oggetti di classe k è dato da:

A(n,k)=n*(n—1)*.....(n-k+1)

Esempio:

se n=3, k=2 si avrà:

$A(3,2)=3*2=6$, infatti con a1,a2,a3 si hanno le seguenti sei disposizioni di due oggetti:

a1,a2 a1,a3 a2,a3
a2,a1 a3,a1 a3,a2

Permutazioni

Si definisce permutazione di n elementi diversi presi r alla volta, i gruppi di r elementi che si possono ottenere con gli n elementi di partenza in maniera che ciascun gruppo sia diverso dagli altri o per un elemento o per l'ordine. In formula la permutazione è data da:

$$P(n,r)=n*(n-1)*......(n-r+1)+n!/(n-r)!$$

in questa formula compare il termine n!(si legge n fattoriale) e viene definito come: $n!=n*(n-1)*(n-2).......1$; si osserva che per definizione $0!=1$.

Esempio: il numero di permutazioni del le lettere a,b,c prese due alla volta è: $P(3,2)=3*2=6$; infatti queste permutazioni sono:

ab,ba,ac,ca,bc,cb

Combinazioni

Le combinazioni di n oggetti diversi presi r alla volta sono i gruppi di r elementi che si possono ottenere dagli n elementi di partenza in modo che ciascun gruppo differisca dagli altri almeno per un elemento. La sua formula è data da:

$$C(n,r)=n!/R!*(n-r)!=P(n,r)/r!$$

Esempio: il numero di combinazioni delle lettere a,b,c, prese due alla volta è:

$$C(3,2)=3*2/2!=3*2/1*2=3$$

in quanto le combinazioni sono solo ab,ac,bc. Infatti, ad esempio, non compare ba in quanto esso differisce da ab solo per l'ordine in cui compaiono le lettere, per la definizione data di combinazione, esse devono differire almeno per un elemento.

IL COMMODORE 64 AL MICROSCOPIO

di **Marco De Rosa**

parte 4ª

(646) - Codice colore del carattere corrente

Questa locazione contiene il codice del colore del carattere attualmente sotto il cursore. Il valore di default è 14 (azzurro chiaro). E' possibile scegliere tra i sedici colori disponibili, forzando i valori tra 0 e 15, il che corrisponde ad usare il comando 'CTRL numero codice'.

(648) - Indirizzo pagina schermo

Questa locazione contiene l'indirizzo del primo byte della pagina schermo, diviso per 256. Il suo valore di default è 4, il che significa che normalmente lo schermo comincia dalla locazione 1024 e arriva alla 2023. E' però possibile forzare questo valore e spostare lo schermo in 64 posti diversi! Provate il programma seguente:

```
10 REM TEXTMOVE
20 S1=4
30 POKE56.32
40 PRINT"+ CLR| QUESTO E' IL PRIMO SCHERMO !"
45 PRINT"PREMI UN TASTO"
50 POKE648,32
55 POKE53272,(PEEK(53272)AND15)OR16
60 PRINT"+ CLR| QUESTO E' IL SECONDO SCHERMO !"
65 PRINT"PREMI UN TASTO"
70 POKE648,4
100 GETA$:IFA$=""THEN100
110 IFS1=4THENS1=32:S2=128:GOTO130
120 S1=4:S2=16
130 POKE648,S1
140 POKE53272,(PEEK(53272)AND15)ORS2
150 GOTO100
```

Ricordate che i due schermi devono essere negli stessi 16K visti dal VIC II.

(649) - Dimensioni Buffer di tastiera

Questa locazione contiene il valore della lunghezza in caratteri del buffer di tastiera. Il suo valore di default è 10, ma può essere portato da 1 a 255. Nel caso voleste disabilitare la tastiera come dispositivo di input, digitate POKE649,0.

(650-651) - Ripetizione automatica sui tasti

Il valore di default di questa locazione è 0, che corrisponde all'auto-repeat disabilitato su tutti i tasti tranne lo spazio, i 4 CRSR e il tasto DEL. Per disabilitarli tutti digitate POKE650,100. Per abilitarli tutti POKE650,255.

La locazione 651 controlla invece il ritardo con cui comincia la ripetizione. Il suo valore di default è 4.

(653) - Pressione tasti di SHIFT

Sul CBM 64 ci sono tre tasti, SHIFT CTRL e COMMODORE, che vanno premuti insieme a qualche cos'altro per avere un effetto. Questa locazione controlla se questi tasti sono premuti o meno. I valori assunti sono:

DEFAULT	0
SHIFT	1
COMMODORE	2
SHIFT+COMMODORE	3
CTRL	4
SHIFT+CTRL	5
COMMODORE+CTRL	6
SHIFT+COMMODORE+CTRL	7

(659-673) - Area RS-232

Queste locazioni sono usate per le comunicazioni attraverso la porta RS232. Ricordate che l'apertura di un file RS232 alloca automaticamente un buffer di 512 bytes che serve a prevenire la perdita di dati in trasmissione o ricezione. Questo viene aperto sotto il tetto del basic indicato nelle locazioni 55 e 56, e può distruggere il programma. Per maggiori informazioni consultate la Programmer's Reference Guide.

(788-789) - Interrupt request routine

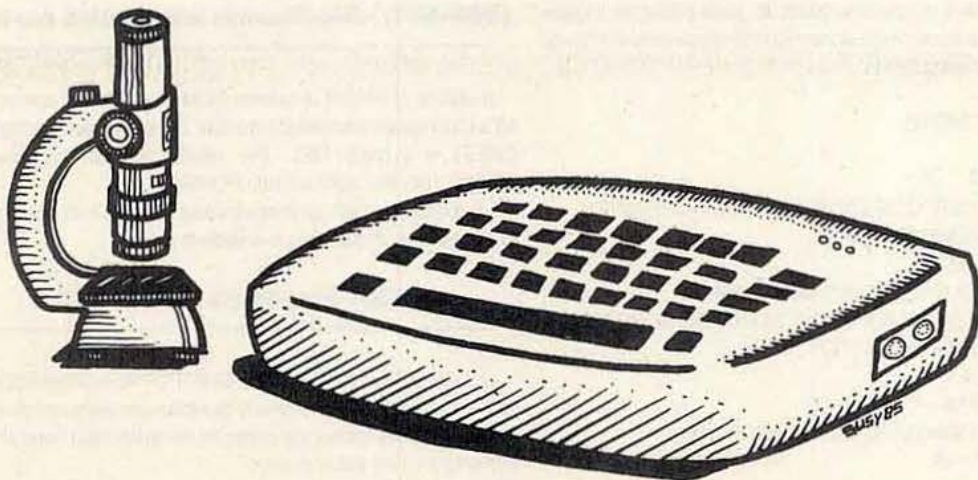
Questa locazione contiene l'indirizzo delle routines di IRQ (interrupt request). Il suo valore di default è 59953. Modificando l'indirizzo è possibile disabilitare il tasto di RUN/STOP. Per disabilitare: POKE788,52. Per riabilitare POKE788,49. Fate attenzione perchè questo disabiliterà anche il clock interno, rendendo impossibile l'uso delle variabili TI e TI$.

Un metodo per disabilitare il tasto di RUN/STOP senza spegnere l'orologio è usare la locazione 808 che vedremo in seguito. Basta comunque digitare POKE808,239. Per riabilitare il tasto POKE808,237. Invece POKE808,225 e POKE808,237 abilitano e disabilitano il tasto RESTORE.

(794-819) KERNAL JUMP TABLE

Il KERNAL è il sistema operativo del CBM 64. Tutte le operazione di input, output, e memory management sono controllate da esso. E' contenuto nella ROM che va da $e000 a $FFFF, cioè da 57344 a 65535 decimale. Tutte le volte che una istruzione BASIC deve essere eseguita, il controllo delle operazioni viene passato al KERNAL che esegue le corrispondenti routines in linguaggio macchina. La KERNAL JUMP TABLE è una tabella in ROM, da 65280 a 65535, che contiene gli indirizzi di 39 delle routines di sistema.

Quando si scrive un programma in linguaggio macchina, è spesso conveniente usare le routines che già fanno parte del sistema operativo, nei casi in cui è possibile, cioè per le operazioni di I/O, orologio ecc. Per utilizzarle basterebbe chiamare l'indirizzo della ROM dove esse effettivamente risiedono. Se però il KERNAL viene modificato in una successiva release della macchina, è molto probabile che l'indirizzo della stessa routine sia cambiato. Questo renderebbbe il programma inutilizzabile. La Commodore ha pensato allora di inserire nell'ultima pagina di memoria una tabella con gli indirizzi delle routnes principali. Invece di chiamare l'indirizzo dove realmente risiede la routine, si può chiamare quello corrispondente della tavola. La Commodore, nelle nuove release di sistema, si occuperà di cambiare la JUMP TABLE e i vostri programmi continueranno a girare.

Ci sono 12 di queste routine che possono essere modificate anche dall'utente. Infatti la JUMP TABLE, nel loro caso, non punta all'indirizzo reale, ma ad un'altra JUMP TABLE residente su RAM in pagina tre, che è poi quella di cui ci occupiamo in questo momento. Gli indirizzi della seconda, essendo su RAM, possono essere modificati dall'utente permettendo così l'inserimento di routines personali. Le locazioni 814 e 815,

contengono addirittura un vettore definibile dall'utente. Il seguente programma di numerazione che le sfrutta è tratto dal libro Master Memory Map di Pavelko e Kelly:

```
63992 INPUT"+CLR| LINEA DI PARTENZA";A
63993 INPUT"INCREMENTO";B:POKE815,B:PRINT
      "+CLR|"
63994 B= A/256:POKE784,(B-INT(B))
      *256:POKE814,B:PRINTA
63995 GETA$:PRINTA$;:IFA$< > CHR$(13)THEN63995
63996 PRINT"GOTO63998":FORA=631TO634:
      POKEA,145:NEXT
63997 POKEA,13:POKE636,13:POKE198,6:END
63998 PRINT"+2UP|":FORA=1TO3:PRINTSPC(9):
      NEXT:PRINT"+3UP|"
63999 A=PEEK(784)+PEEK(814)
      *256+PEEK(815):GOTO63994
```

(1024 - 2023) Area schermo

Questa è la zona dove è memorizzato il contenuto dello schermo. Per esempio, per stampare in alto a sinistra una A basta scrivere POKE1024,1. Nei primi esemplari del CBM 64, ogni carattere forzato sullo schermo in questo modo veniva scritto in bianco. Negli ultimi modelli invece il colore è uguale a quello dello schermo, cioè blu. Per rendere visibile il carattere è necessario digitare POKE55296,1. Lo schermo ha infatti un corrispondente per quanto riguarda il colore situato dalla locazione 55296 al 56295. Potete quindi indirizzare il colore di ogni singolo carattere dello schermo. Potete vedere qui di seguito i codici colore:

NERO	0
BIANCO	1
ROSSO	2
BLU VERDE	3
PORPORA	4
VERDE	5
BLU	6
GIALLO	7
ARANCIO	8
MARRONE	9
ROSSO CHIARO	10
GRIGIO 1	11
GRIGIO 2	12
VERDE CHIARO	13
AZZURRO	14
GRIGIO 3	15

Puntatori agli Strites

Queste otto locazioni contengono il numero del blocco dove sono memorizzati rispettivamente gli otto sprites. Il numero da inserire nelle locazioni, da 0 a 255, si calcola dividendo l'indirizzo di memorizzazione per 64.

HEX	DEC	DESCRIZIONE
0285	645	Flag: variabile Kernal per Time Out IEEE.
0286	**646**	**Codice colore del carattere corrente.**
0287	647	Codice colore sfondo cursore.
0288	**648**	**Pointer: inizio memoria schermo.**
0289	**649**	**Dimensioni buffer di tastiera.**
028A	**650**	**Flag: ripetizione automatica sui tasti. Il suo valore di default è zero.**
028B	**651**	**Contatore velocità ripetizione tasti. Il suo valore di default è 4.**

HEX	DEC	DESCRIZIONE
028C	652	Contatore ritardo ripetizione tasti.
028D	**653**	**Flag: pressione tasti SHIFT, CTRL e COMMODORE.**
028E	654	Ultima configurazione di SHIFT.
028F-0290	655-656	Vector: setup della tabella di tastiera. Il suo valore di default è 60232.
0291	657	Flag: 0=tasto COMMODORE disabilitato, 128=abilitato.
0292	658	Flag: 0=scroll automatico verso il basso abilitato, Altro valore=disabilitato.
0293	**659**	**RS232: immagine del registro di controllo del 6551.**
0294	**660**	**RS232: immagine del registro di comando del 6551.**
0295-0296	**661-662**	**RS232: BPS non standard.**
0297	**663**	**RS232: registro di stato per RS232.**
0298	**664**	**RS232: Numero di bits rimasti da spedire.**
0299-029A	**665-666**	**RS232: Baud Rate-Bit in microsecondi**
029B	**667**	**RS232: indicatore RS232 al termine del buffer di input**
029C	**668**	**RS232: inizio del buffer di input.**
029D	**669**	**RS232: inizio del buffer di output.**
029E	**670**	**RS232: indice al termine del buffer di uscita.**
029F-02A0	671-672	Immagazzinamento temporaneo di IRQ durante la lettura del nastro.
02A1	**673**	**Abilitazione RS232.**
02A2	674	Controllo sensore cassetta durante le operazioni di I/O.
02A3	675	Area temporanea per cassetta.
02A4	676	Indicatore temporaneo di IRQ durante la lettura del nastro.
02A5	677	Indicatore temporaneo per indice linea di schermo.
02A6	678	Flag: sistema di modulazione.0=NTSC, 1=PAL.
02A7-02FF	679-767	Non usati. Area utile per la memorizzazione di sprites.
0300-0301	768-769	Vector: Stampa messaggi errore del BASIC.
0302-0303	770-771	Vector: Warm Start del BASIC.
0304-0305	772-773	Vector: routine di Tokenize del BASIC.

HEX	DEC	DESCRIZIONE
0306-0307	774-775	Vector: LIST del testo BASIC. POKE 775,200 disabilita il comando LIST. Per la riabilitazione digitate POKE 775,167.
0308-0309	776-777	Vector: indicatore ingresso indiretto per la ricerca delle parole chiave in BASIC. Il valore di default è $A7E4.
030A-030B	778-779	Vector: riconoscimento delle tokens del BASIC.
030C	780	Immagine dell'accumulatore del 6502.
030D	781	Immagine del registro X del 6502.
030E	782	Immagine del registro Y del 6502.
030F	783	Immagine dello Stack Pointer del 6502.
0310	784	Contiene il codice dell'istruzione macchina JMP, cioè 76.
0311-0312	785-786	Pointer: indirizzo di partenza della routine USR(X).
0313	787	Non usato.
0314-0315	**788-789**	**Vector: Interrupt Request Routine. Il valore di default è 59953.**
0316-0317	790-791	Vector: Interrupt per l'istruzione BRK.
0318-0319	792-793	Vector: Interrupt non mascherabile. Il valore di default è 65095.
031A-031B	**794-795**	**Vector: routine di OPEN del Kernal.**
031C-031D	**796-797**	**Vector: routine di CLOSE del Kernal.**
031E-031F	**798-799**	**Vector: routine di CHKIN del Kernal.**
0320-0321	**800-801**	**Vector: routine di CHKOUT del Kernal.**
0322-0323	**802-803**	**Vector: routine di CLRCHN del Kernal.**
0324-0325	**804-805**	**Vector: routine di CHRIN del Kernal.**
0326-0327	**806-807**	**Vector: routine di CHROUT del Kernal.**
0328-0329	**808-809**	**Vector: routine di STOP del Kernal.**
032A-032B	**810-911**	**Vector: routine di GETIN del Kernal.**
032C-032D	**812-813**	**Vector: routine di CLALL del Kernal.**
032E-032F	**814-815**	**Vettore definibile dall'utente.**
0330-0331	**816-817**	**Vector: routine di LOAD del Kernal.**
0331-0333	**818-819**	**Vector: routine di SAVE del Kernal.**
0334-033B	820-827	Non usato.
033C-03FB	828-1019	Buffer di I/O del registratore. E' utile per memorizzare degli sprites.
03FC-03FF	1020-1023	Non usati.
0400-07E7	**1024-2023**	**Area schermo.**
07F8-07FF	**2040-2047**	**Puntatori agli sprites.**

PROGRAMMAZIONE STRUTTURATA

di **Mariangela Guardione**

Quinta parte

In questa puntata si prenderanno in considerazione i files di dati, che rappresentano una struttura molto importante nella programmazione. Il primo tipo di files che tratteremo sono quelli associati alle unità periferiche, lo strumento con cui il computer comunica con l'esterno: la tastiera/schermo-video e la stampante. Per immettere o ricevere attraverso il computer è infatti necessario che i dati siano acquisiti o emessi in forma di sequenza di caratteri. L'interprete Basic deve essere in grado di comprendere, a seconda del tipo di variabili coinvolte, se sia necessario eseguire delle opportune conversioni (che portino ad avere i dati numerici emessi nella rappresentazione atta ad eseguire calcoli algebrici) oppure se i dati numerici emessi devono essere visualizzati in una forma leggibile dall'utente.

Questo insieme d'informazioni, che proviene oppure che è emesso da un'unità periferica, prende il nome di "file".

Caratteri riconosciuti dal Basic

L'interprete Basic riconosce i caratteri seguenti.

- alfabeto inglese (maiuscole e minuscole)
- numeri (0,1,2,3,4,5,6,7,8,9)

Caratteri speciali:

- Spazio bianco
- ; Punto e virgola
- = Segno di uguale o simbolo di assegnazione
- + Segno più
- - Segno meno
- * Simbolo per l'operazione di moltiplicazione
- / Simbolo per l'operazione di divisione
-] Simbolo per l'operazione di elevamento a potenza
- (Parentesi sinistra (aperta)
-) Parentesi destra (chiusa)
- % Simbolo di percentuale
- ≠ Simbolo di cancelletto o pound
- $ Segno di dollaro
- ! Punto esclamativo
- , Virgola
- . Punto o punto decimale
- : Due punti
- & E commerciale
- ? Punto interrogativo
- < Minore di
- > Maggiore di
- @ Chiocciolina

A questo punto ecco descritte, una per una, tutte le istruzioni e le funzioni che permettono al linguaggio Basic di comunicare con l'esterno tramite le periferiche.

Istruzioni di ingresso/uscita

- **Funzione Get.** Permette di eseguire un tentativo di lettura. Infatti, se in quel momento un tasto era premuto, viene restituito il carattere letto. In caso contrario viene ritornata la stringa nulla (vedi figura 1)

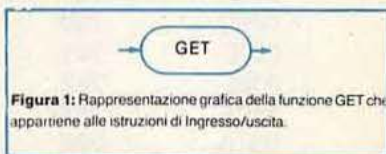



Figura 1: Rappresentazione grafica della funzione GET che appartiene alle istruzioni di Ingresso/uscita.

- **Istruzione Input.** La struttura in pseud o – codifica di questa istruzione è rappresentata nella figura 2.

Con questa istruzione il programma si ferma e compare sul video un punto interrogativo che indica un'attesa di dati. I dati introdotti sono, oltre ad essere visualizzati sullo schermo, assegnati alla (o alle) variabile (variabili) della lista di nomi.

Ciascuno di questi elementi sono separati fra di loro da una virgola e il loro numero deve essere uguale a quello relativo ai "nomi di variabile" che sono presenti nella lista e per concludere l'introduzione di questi si deve premere il

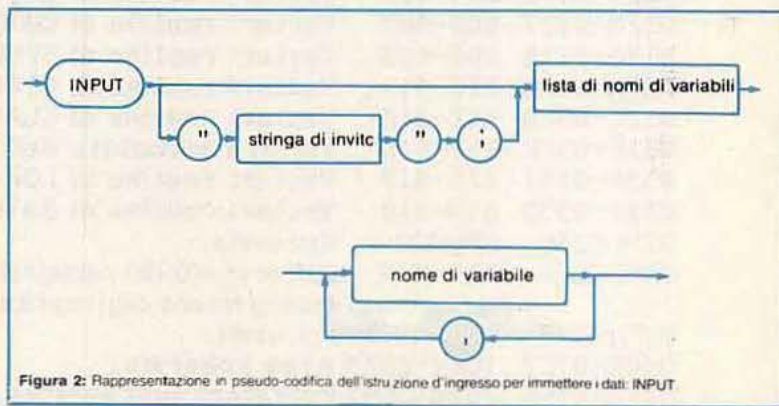



Figura 2: Rappresentazione in pseudo-codifica dell'istru zione d'ingresso per immettere i dati: INPUT

tasto «return». Dopo che sono stati inseriti i dati all'interno del computer, le sequenze di caratteri sono convertite automaticamente nel formato binario che rappresenta il linguaggio tipico del computer.

• **Istruzione Print.** La rappresentazione in pseudo-codifica di questa istruzione è data dalla figura 3. Serve per visualizzare i dati di una elaborazione su schermo o, se seguita dall'indirizzo della device (≠n) successivo a una open del canale logico relativo, sulla stampante.

L'utilizzo della virgola nel Basic permette di suddividere la linea di stampa in "zone", mentre il punto e virgola opera in modo che il prossimo valore della lista di espressioni sia emesso immediatamente dopo l'ultimo valore stampato.

Nel caso in cui la lunghezza della linea da stampare sia maggiore dell'ampiezza della linea fisica, il Basic continua a stampare andando automaticamente all'inizio della linea fisica seguente.

In fase di stampa si ha: il numero è sempre seguito da uno spazio, i numeri positivi sono preceduti da uno spazio, i numeri negativi sono preceduti dal segno meno.

Funzioni per il formato stampa

• **Tab.** Questa funzione esegue la stampa a partire da una determinata posizione i-esima. Se la posizione di stampa è già oltre la i-esima, Tab fa andare alla posizione indicata ma sulla linea seguente.

La rappresentazione in pseudo-codifica di questa istruzione è data nella figura 4.

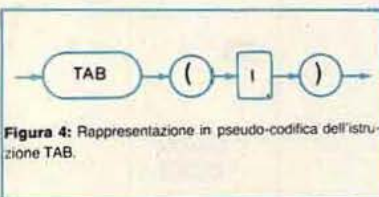



Figura 4: Rappresentazione in pseudo-codifica dell'istruzione TAB.

Un esempio:
Print Tab(20);"*" stampa il simbolo asterisco in colonna 20

• **Spc.** Questa istruzione invece permette di eseguire la stampa iniziando dalla posizione attuale del cursore e si sposta del numero di spazi che vengono indicati.

La sua rappresentazione in pseudo – codifica è indicata in figura 5. Facciamo un esempio:

Print Spc(20);"*";Spc(10);"*" stampa un'asterisco nella colonna 20 e un'altro nella colonna 31 in quanto l'istruzione Spc comincia a contare dallo spazio immediatamente dopo il primo asterisco e stampa l'altro dopo dieci posizioni.

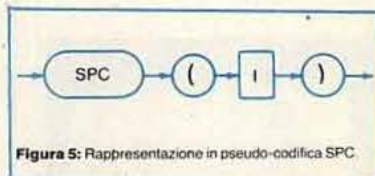



Figura 5: Rappresentazione in pseudo-codifica SPC

Figura 3: Rappresentazione in pseudo-codifica dell'istruzione per l'emissione dei dati: PRINT

A questo punto prendiamo in considerazione come avviene la trasmissione dei caratteri di controllo sui dispositivi atti a rappresentare i risultati di elaborazioni. Tutto questo, come abbiamo già visto, avviene utilizzando le istruzioni Print che inviano sul video o sulla stampante (print≠) i risultati. Se non viene specificato un "punto e virgola" come ultimo carattere, il cursore si posiziona automaticamente all'inzio della riga successiva, in quanto il Basic, dopo tutti i caratteri che rappresentano il dato da emettere aggiunge i caratteri relativi al codice ASCII che serve per portarsi alla linea fisica successiva.

Questi caratteri di controllo sono inviati al dispositivo periferico utilizzando espressioni stringa con un'istruzione Print. Tutti questi caratteri sono trasmessi utilizzando la codifica ASCII come argomento della funzione CHR$.

I più comuni caratteri di controllo per il C64 sono riportati nella tabella 6 (presente nel manuale fornito con la macchina).

A tutto questo sono legate anche altre istruzioni che permettono una specifica dei formati di stampa e che rivestono molta importanza a livello di rappresentazione dei dati (l'utente deve essere in grado di estrarre molto rapidamente da un tabulato le informazioni che più lo interessano).

Si devono cioè combinare le istruzioni Spc, di gestione stringa e, eventualmente, di trasformazione da numero in stringa, in maniera da ottenere dei campi di

stampa ben definiti onde identificare le posizioni in cui si vuole che siano scritte le cifre dei numeri di un prospetto e le relative diciture.

In altre parole il File rappresenta l'organizzazione con cui le informazioni sono immagazzinate su supporto cartaceo negli archivi. Cioè: un dato elementare, il

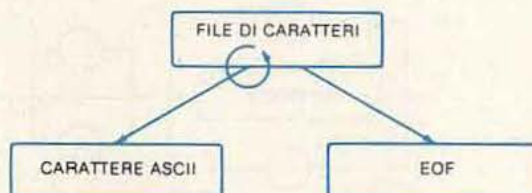
STAMPA	CHR$	STAMPA	CHR$	STAMPA	CHR$	STAMPA	CHR$
	0	CRSR	17	f1	133		155
	1	RVS ON	18	f3	134	PUR	156
	2	CLR HOME	19	f5	135	CRSR	157
	3	INST DEL	20	f7	136	YEL	158
	4		21	f2	137	CYN	159
WHT	5		22	f4	138	SPACE	160
	6		23	f6	139		
	7		24	f8	140		
disabilita SHIFT C=	8		25	SHIFT RETURN	141		
abilita SHIFT C=	9		26	PASSAGGIO AL MAIUSCOLO	142		
	10		27		143		
	11	RED	28	BLK	144		
	12	CRSR	29	CRSR	145		
RETURN	13	GRN	30	RVS OFF	146		
PASSAGGIO IN NEGATIVO	14	BLU	31	CLR HOME	147		
	15	SPACE	32	INST DEL	148		
	16						

Figura 6: I caratteri di controllo del C64

Files di caratteri ad accesso sequenziale

Un file è formato da una sequenza di caratteri. Un'infomazione può essere scritta solo di seguito a quella registrata nell'operazione precedente di scrittura. Vedi anche la figura 7.

byte, viene scritto in una casella (il campo), su una scheda il record, che insieme a tutte le altre simili viene contenuta in un classificatore che è il File (vedi figura 9).

Quando si ha un programma che si vuole trasferire dalla memoria centrale ad un'unità di memoria si esegue infatti l'operazione di Save su di un nastro o di

un dischetto. Per richiamarlo in memoria si deve far eseguire l'istruzione Load, con l'avvertenza però che ogni programma abbia un nome distinto che permette al computer di identificarlo. Al file è legata la seguente operazione:
r - Lettura del file di dati

della scrittura l'istruzione close riveste un ruolo importante in quanto serve per chiudere, dopo aver terminato l'accesso, il file logico, operazione che evita la perdita dei dati appena inseriti. La sua rappresentazione grafica è data dalla Fig.11

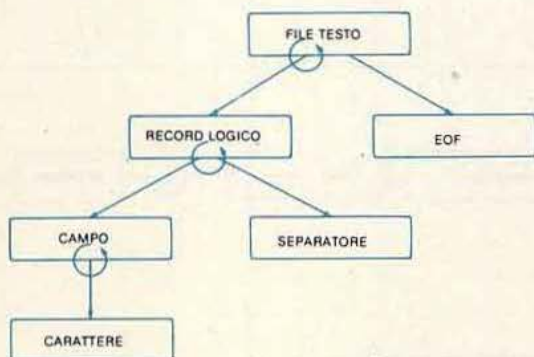



Figura 7a: Rappresentazione logica del FILE SEQUENZIALE.

Per poter prelevare dati dai supporti periferici, nastri o dischetti, è necessario aprire un canale di lettura utilizzando l'istruzione Open. Questa istruzione permette anche di:

- Definire le caratteristiche del file e il tipo di accesso che è consentito e tutto questo serve per definire le modalità con cui possono operare le istruzioni di lettura o di scrittura.
- Creare un'associazione fra il nome rappresentativo con cui sia riconosciuto il file all'interno del programma, e che viene scelto dal programmatore, e il nome effettivo con cui il file è archiviato all'interno della directory, che rappresenta una tabella indicante per ogni file quali blocchi sono occupati.

La rappresentazione grafica del file, la sua apertura e chiusura è data dalla Fig.9, mentre per quanto riguarda la lett ura la sua rappresentazione grafica è esplicata nella Fig.10.

Per la scrittura valgono le stesse considerazioni fatte per la lettura. Nel caso

Il programma

A conclusione, come di consueto, ecco il listato del programma che gestisce le schedine di Pubblica Sicurezza che ogni albergo deve tenere per legge.

Con questo strumento si può avere, quindi, a disposizione la situazione aggiornata delle presenze nonchè i dati anagrafici relativi ad ogni cliente. Ogni dato può essere, inoltre, variato in caso di errore o necessità.

Da notare che viene usato un file di dati (per la memorizzazione permanente) ed una matrice per ricopiare il file ed effettuare tutte le operazioni con notevole guadagno di tempo. In caso di mancanza del file il programma è strutturato in maniera da gestire il messaggio di errore relativo.

Passiamo ora ad esaminare più in dettaglio la sua struttura. Dalla linea **90** alla **250** sono allocate l'apertura del file logico per il test d'errore, il dimensionamento delle matrici e **i** richiami alle su-

broutines d'intestazione e di test sull'esistenza del file dati.

Dalla **350** alla **1100** si trovano le assegnazioni stringa usate nel programma, dalla **1250** alla **1800** vi è la routine relativa alla maschera di scelta.

Dalla **1900** alla **3350** è allocata la routine relativa all'arrivo clienti, mentre dalla **3400** alla **3650** si trova una subroutine utilizzata nella routine posta. Dalla **3700** alla **5150** relativa alla partenza dei clienti, dalla **5200** alla **5400** è allocata la subroutine utilizzata nella routine posta.

Dalla **5450** alla **7000** si permette la ricerca mirata ed eventuali variazioni sui dati, mentre dalla **8700** alla **9450** è allocata la routine di chiusura con scrittura dei dati presenti nella matrice sul file.

Dalla **9500** alla **9950** si trova la routine di lettura del file dati contenente il test di errore (**9510**; **9520**; **9720**)

Dalla **10000** alla **10600** si trova la routine per la maschera iniziale mentre dalla **11250** alla **12500** sono allocate alcune routines di gestione video

Dalla **13000** alla **13040** si trova un'ulteriore routine di test d'errore.

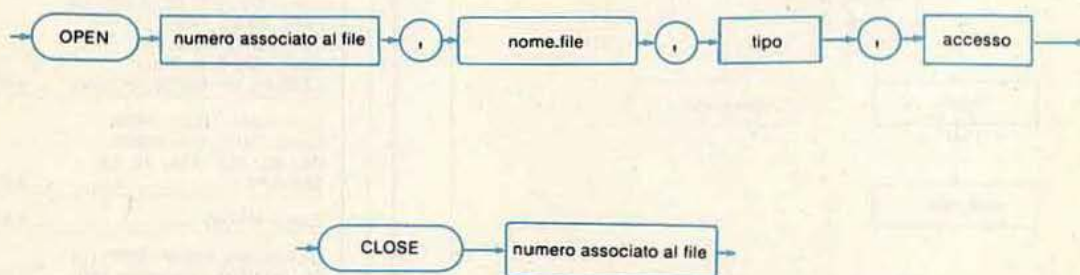



Figura 8: Rappresentazione in pseudo-codifica dell'apertura e chiusura di un file sequenziale per il C64

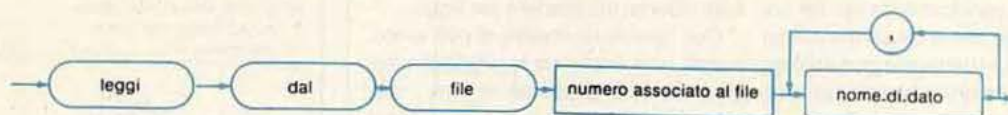



Figura 9: Rappresentazione in pseudo-codifica dell'opera zione di lettura di un file sequenziale.

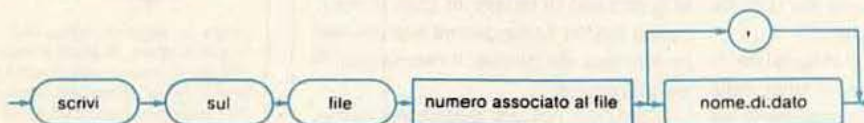
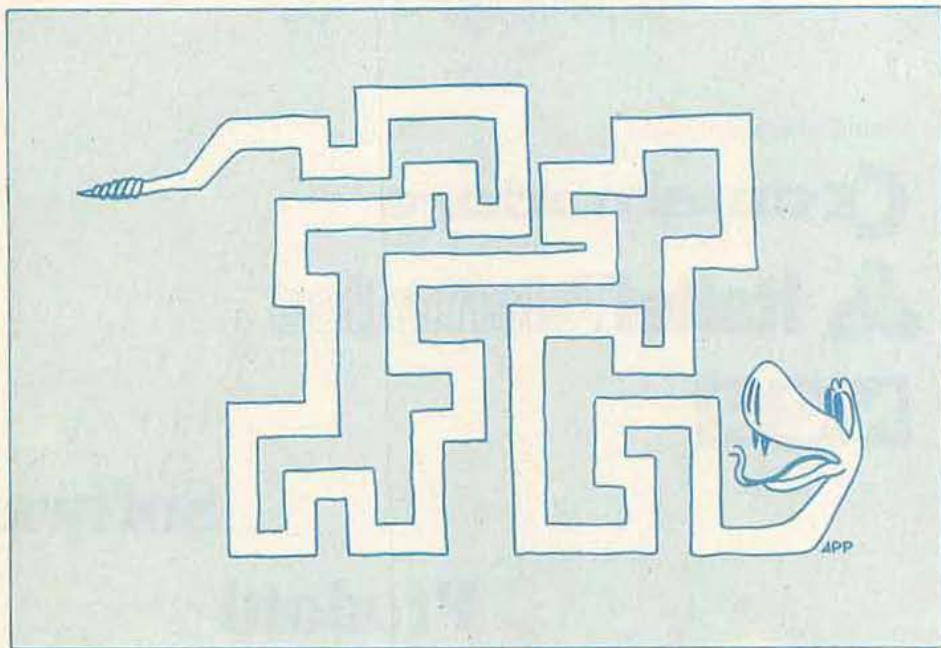



Figura 10: Rappresentazione in pseudo-codifica dell'ope razione di acrittura di un file sequenziale.

Il filo di Arianna

di **Marco De Rosa e Sandro Sorgi**

"Un labirinto ingannatore così ingegnosamente concepito non fu mai visto al mondo, né prima né da allora in poi. Non può esservi null'altro di così intricato, salvo la mente di un uomo come Dedalo che lo inventò, o il cuore di un qualsiasi uomo normale...".

da *"Tanglewood Tales"*
di Nataniel Hawthorne

"Labirinto: luogo, edificio, intreccio di strade o passaggi dove è difficile orizzontarsi o da cui è difficile uscire. Gioco di pazienza che consiste nel trovare, tra tante strade tortuose disegnate, l'unica che conduce all'uscita".

dal vocabolario Zingarelli

Se consideriamo un labirinto dal punto di vista strettamente matematico, esso diventa un problema della branca di questa scienza che studia le proprietà delle superfici: la topologia.

I labirinti si dividono in due gruppi principali: semplicemente connessi e molteplicemente connessi. I primi hanno la seguente proprietà: per ogni coppia di punti all'intern t del labirinto, esiste uno ed un solo percorso di unione. Questo equivale a dire che essi non contengono circuiti chiusi (loop), o pareti isolate (fig. 5). I secondi invece possono avere ambedue le caratteristiche (una è conseguenza dell'altra, come potete uadere in fig. 6). Esistono dei semplici algo ritmi da la risoluzione dei labirinti, Nel caso di quelli semplicemene connessi è sufficiente tenere la mano sul muro destro (o sinistro) durante tutto il percorso. La strada vi condurrà sicuramente all'uscita, anche se non secondo la via più breve. Per quelli molteplicemente connessi la cosa è leggermente più complessa. Una descrizione completa è data in *"Ricreazioni matematiche"* di Eduard Lucas, ed è la seguente:

1/ Entrando si tracci una linea lungo la parte destra (o sinistra) del percorso. Questo corrisponde a tenere la mano appoggiata alla parete.

2/ Se si arriva in un punto in cui si incrociano due o più strade, si scelga casualmente una di esse.

3/ Se percorrendo una strada nuova si arriva in un punto in cui si incrociano due o più strade, o in un vicolo cieco e si è gia passati in quel punto, si gira e si torna indietro.

4/ Se arriva in un punto in cui si inrociano due o più strade e si è già passati in quel punto, si prenda un percorso nuovo, se esiste. Altrimenti se ne prenda uno vecchio.

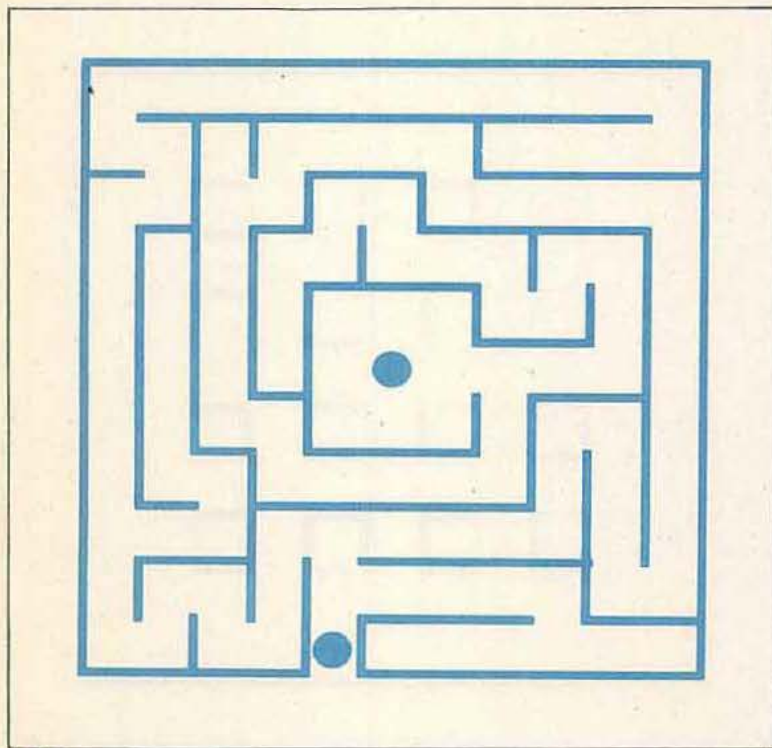



Fig. 5

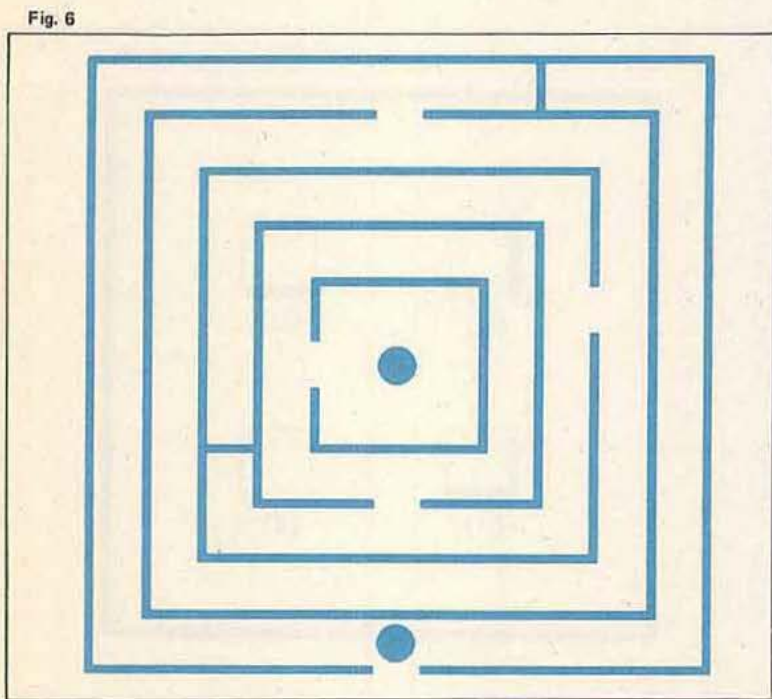



Fig. 6

5/ Non prendere mai una strada segnata da entrambi i lati.

Il programma

Dato il consueto RUN al programma, comparirà un menù con quattro scelte:
1/ **GENERA**: è il cuore del programma. Una volta inseriti il numero di righe e colonne del labirinto (massimo 38 x 21) partirà la generazione. Alla fine potrete stampare e salvare il labirinto. Al nome del file verrà aggiunto il prefisso "LAB.".
2/ **CARICA**: permette di ricaricare dal disco dei labirinti precedentemente salvati. Al termine della visualizzazione è possibile stampare gli stessi.
3/ **LISTA**: mostra sullo schermo la Directory del dischetto presente nell'unità disco. Sono mostrati soltanto i file di tipo "LÀB.".
4/ **FINE**: permette di uscire dal programma. Lo stesso viene cancellato dalla memoria.

L'algoritmo

Esistono diversi algoritmi per la generazione dei labirinti, quello che vi proponiamo, oltre ad essere originale, è anche incredibilmente corto. Nel listato occupa le linee dalla 440 alla 540, e avrebbe potuto essere ulteriormente compattato se non avessimo dovuto simulare l'istruzione PRINTATX,Y assente nel Basic del C-16.

Le sedici possibili celle visibili in fig. 1 sono ridotte alle 3 di figura 2 se consideriamo il labirinto illuminato da una luce fittizia proveniente dall'angolo di Nord-Est. In questo modo i muri a Nord e a Est non sono visibili, e verranno costruiti in seguito dalle celle generate in quelle direzioni.
L'idea è la seguente:
1/ Si parte con un "seme" nella cella all'estremo Sud-Ovest. Abbiamo scelto per comodità la cella (c) di fig. 1.
2/ Si comincia a generare casualmente una pista usando le tre possibili celle, finche non ci si trova nell'impossibilità di proseguire. A questo punto la pista viene chiusa.
3/ Si comincia allora a scandire il labirinto finchè non si trova una cella ancora

vuota. In questo caso ci si attacca alla cella precedente, sicuramente piena, e da questa si riparte a costruire una nuova pista.

Se si segue alla lettera questo algoritmo, ci si ritrova con un labirinto semplicemente connesso, della forma rettangolare di dimensioni prescelte, senza nessuna zona libera.

Se dal punto due si elimina la parte relativa alla chiusura della pista, si ottiene un labirinto molteplicemente connesso.

Se alla fine della generazione si inseriscono casualmente delle celle vuote, cioè di tipo (d), in punti che non siano già di quel tipo, si ottiene un labirinto molteplicemente connesso, con tanti loop quante sono le celle vuote introdotte.

Se nella matrice S%(40,23) si riempiono delle zone in modo che l'algoritmo pensi che siano già state occupate con delle celle, si possono ottenere labirinti di qualsiasi forma, sempre contenuti all'interno del rettangolo di dimensioni prescelte.

Le variabili

S%(40,23): è isomorfa al labirinto. All'inizio è riempita di zero (cella vuota), tranne che nei bordi del labirinto che vengono fissati a 1. Durante la generazione, ogni elemento viene riempito con il codice schermo del carattere corrispondente alla cella generata.

A(4,4): contiene i codici schermo dei caratteri corrispondenti alla cella da visualizzare, in funzione della direzione di provenienza (OD), e della direzione in cui si proseguirà (D). Il codice 42 non viene mai stampato se l'algoritmo opera correttamnete. Vedi esempio in fig. 3.

D: direzione di uscita della cella generata. 1=nord, 2=ovest, 3=sud, 4=est.

OD: direzione di uscita della cella precedente a quella generata.

F(4): è simile ad A(4,4), ma serve a chiudere la strada. Vedi esempio in fig. 4.

P(4,6): contiene le possibili direzioni in cui può continuare una generazione, in funzione della direzione di provenienza OD.

X,Y: coordinate dell'ultima cella

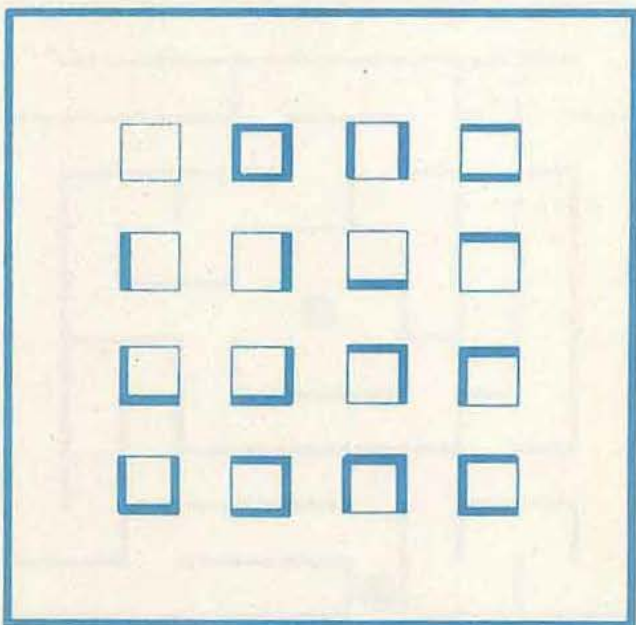



Fig. 1

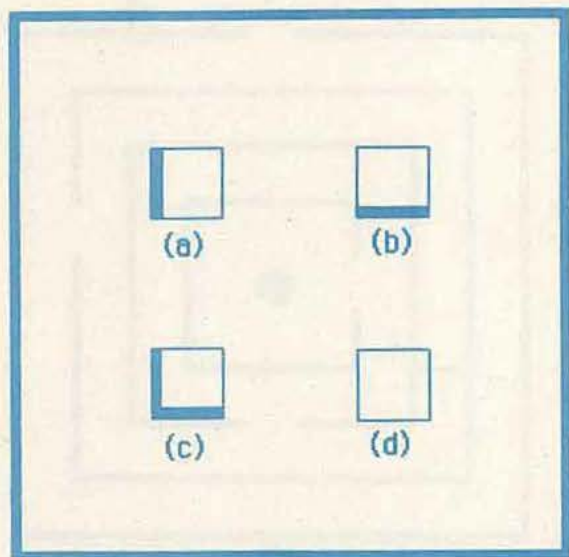



Fig. 2

generata.

NX,NY: coordinate della casella dove si cercherà di generare una nuova cella.

Tutte le altre sono variabili d'appoggio, di loop, o comunque non importanti ai fini dell'esecuzione dell'algoritmo.

Descrizione dell'algoritmo

410: inizializza le variabili e stampa i contorni a nord e est.

420-430: mette il seme nell'angolo a Sud-Ovest del labirinto.

440: inizio del loop di scansione dello schermo.

450: se la cella S%(I,J) è ancora vuota prosegue, altrimenti continua dopo l'istruzione di loop alla riga 530, cioè ricomincia a scandire in cerca di celle vuote.

460: tenta una direzione di partenza della nuova pista.

470: inizio del loop di ricerca delle possibili direzioni di uscita.1

480-500: tenta un direzione. Se non è possibile ritenta in modo ciclico. Dopo tre tentativi esce alla riga 520.

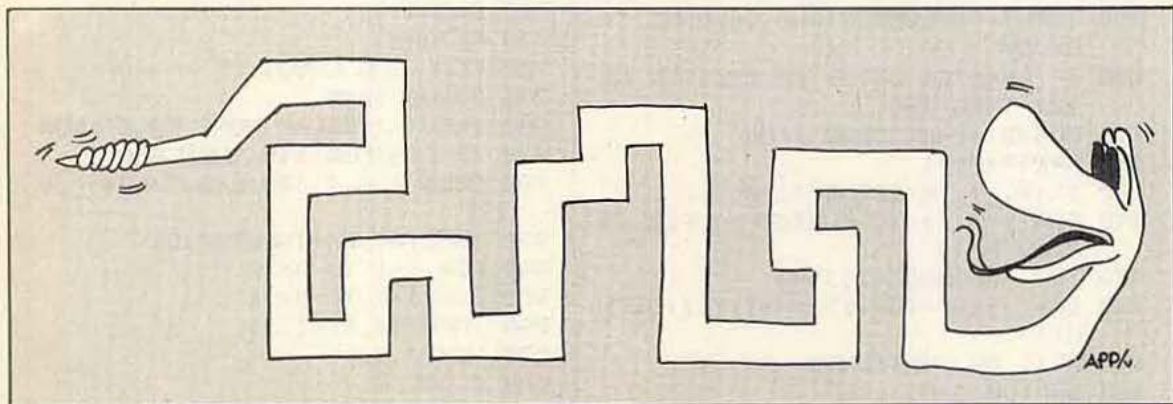
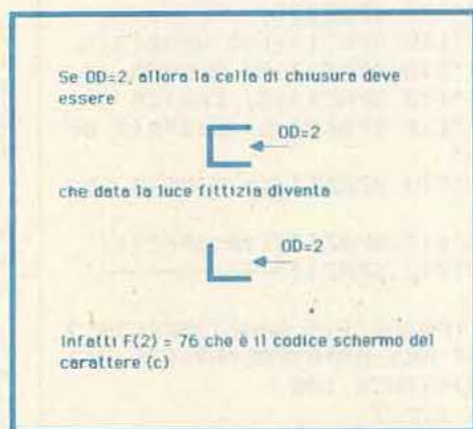
510: fine del loop di ricerca.

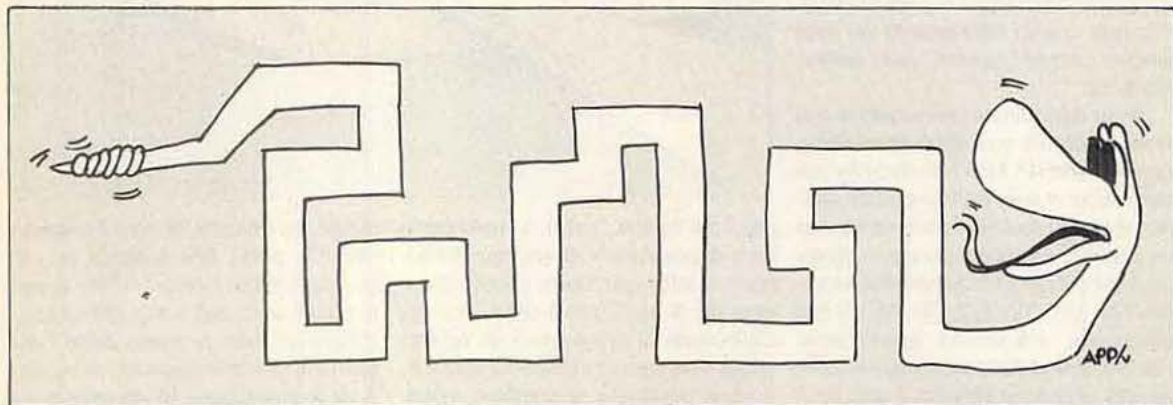
520: chiude la pista.

530: fine del loop iniziato alla riga 450.

540: fine del loop di scansione schermo.

Fig. 3

Fig. 4

```
100 REM  *******************
110 REM  *     ARIANNA     *
120 REM  *                 *
130 REM  * DI MARCO DE ROSA *
140 REM  *  E SANDRO SORGI  *
150 REM  *******************
160 REM  * COMMODORE 16     *
170 REM  * COMMODORE PLUS 4 *
180 REM  *******************
190 REM  * MPS 802          *
200 REM  *******************
210 COLOR 0,1,7:COLOR 4,1,7:COLOR
    1,8,7
220 PRINT"[CLEAR][GIALLO][16 SPAZI
    ]ARIANNA"
230 PRINT:PRINT"[7 SPAZI]IL GENERA
    TORE DI LABIRINTI"
240 PRINT"[4 SPAZI]DI [VERDE2]MARC
    O DE ROSA[GIALLO] E [VERDE2]SA
    NDRO SORGI[GIALLO]"
250 PRINT:PRINT:PRINT:PRINT
260 PRINT"[13 SPAZI]┌──────────┐"
270 PRINT"[13 SPAZI]|[10 SPAZI]|"
280 PRINT"[13 SPAZI]|1) GENERA |"
290 PRINT"[13 SPAZI]|2) CARICA |"
300 PRINT"[13 SPAZI]|3) LISTA[2 SP
    AZI]|"
310 PRINT"[13 SPAZI]|4) FINE[3 SPA
    ZI]|"
320 PRINT"[13 SPAZI]|[10 SPAZI]|"
330 PRINT"[13 SPAZI]└──────────┘"
340 PRINT
350 PRINT:PRINT"[14 SPAZI]SCELTA ?
    ";:GET KEY A$:A=VAL(A$):IF A<1
     OR A>4 THEN 100
360 COLOR 1,2,7
370 ON AGOSUB 390,780,1380,950
380  RUN
390 INPUT "[CLEAR]RIGHE,COLONNE ";
    DX,DY
400 IF DX<1 OR DX>38 OR DY<1 OR DY
    >21 THEN 390
410 GOSUB 1240:GOSUB 1190
420 X=2:Y=DY+1
430 S%(X,Y)=76:GOSUB 1170
440 FOR J=DY+1 TO 2 STEP -1:FOR I=
    2 TO DX+1
450  DO  USINGS%(I,J)=0
460 OD=1-3*(J=(DY+1)):X=I:Y=J:O=FN
    O(1)
470 N=1: DO  UNTILN>3
480 D=P(OD,O+N)
490 NX=X+(D=2)-(D=4):NY=Y+(D=1)-(D
    =3)
500 IF S%(NX,NY)=0 THEN O=FNO(1):S
    %(X,Y)=A(OD,D):GOSUB 1170:X=NX
    :Y=NY:OD=D:N=0
510 N=N+1: LOOP
520 S%(X,Y)=F(OD):GOSUB 1170
530  LOOP
540 NEXTI,J
550 I=2:J=2: DO  UNTIL S%(I,J)=96
    OR I>DX+2
560 J=J+1:IF J=DY+2 THEN I=I+1:J=2
570  LOOP
580 IF S%(I,J)=96 THEN X=I:Y=J:S%(
    X,Y)=81:GOSUB 1170
590 I=DX+2:J=2: DO  UNTIL S%(I,J)=
    96 OR I<2
600 J=J+1:IF J=DY+2 THEN I=I-1:J=2
610  LOOP
620 IF S%(I,J)=96 THEN X=I:Y=J:S%(
    X,Y)=81:GOSUB 1170
630 GOSUB 960
640 GOSUB 660
650 RETURN
660 REM  SUB SALVA
670 PRINT"[HOME][24 DOWN]SALVI[2 S
    PAZI]?";:GET KEY A$:IF A$<>"S"
     THEN RETURN
680 INPUT "[CLEAR]NOME FILE";NO$
690 IF LEN(NO$)>10 THEN 680
700 OPEN 4,8,4,"@0:LAB."+NO$+",S,W
    "
710 PRINT#4,DX:PRINT#4,DY
720 FOR I=1 TO DX+2
730 FOR J=1 TO DY+2
740 PRINT#4,S%(I,J)
750 NEXTJ,I
760 CLOSE 4
770 RETURN
780 REM  SUB LEGGI
790 GOSUB 1240
800 INPUT "[CLEAR]NOME FILE";NO$
810 IF LEN(NO$)>10 THEN 680
820 OPEN 4,8,4,"@0:LAB."+NO$+",S,R
    "
830 INPUT#4,DX:INPUT#4,DY
840 FOR I=1 TO DX+2
850 FOR J=1 TO DY+2
860 INPUT#4,S%(I,J)
870 NEXTJ,I
880 CLOSE 4
```

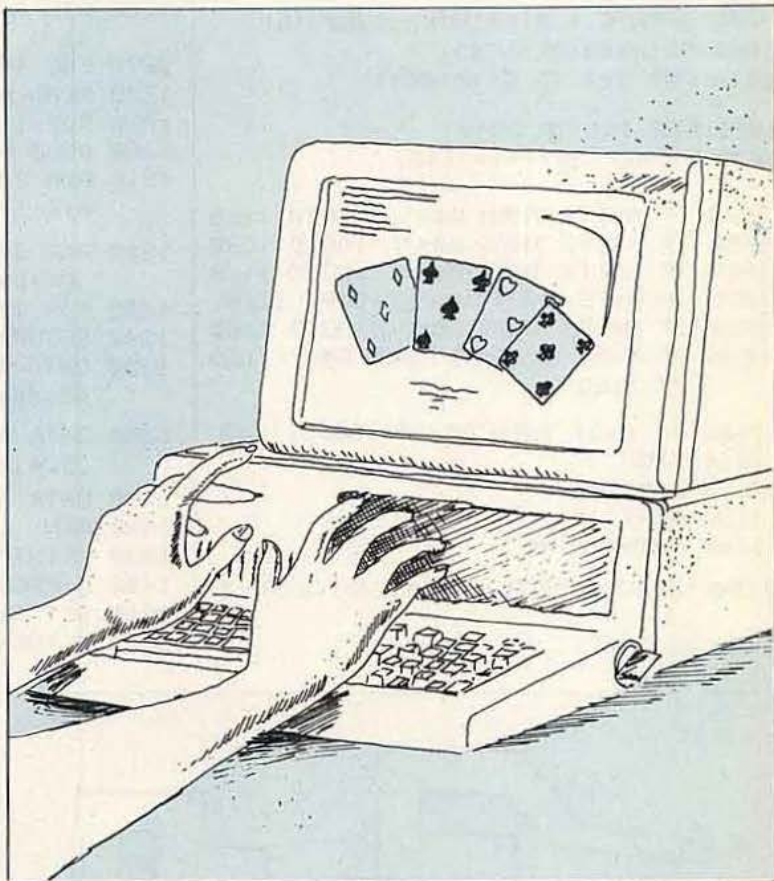
```
 890 GOSUB 1190
 900 FOR X=2 TO DX+1:FOR Y=2 TO DY+
     1
 910 GOSUB 1170
 920 NEXTY,X
 930 GOSUB 960
 940 RETURN
 950 NEW :END
 960 REM  STAMPA
 970 PRINT"[HOME][24 DOWN]STAMPA ?"
     ;:GET KEY A$:IF A$<>"S" THEN R
     ETURN
 980 OPEN 3,4
 990 OPEN 6,4,6:PRINT#6,CHR$(18)
1000 PRINT#3:PRINT#3
1010 FOR J=1 TO DY+2:A$=""
1020 FOR I=1 TO DX+2
1030 A=PEEK(3071+I+J*40)
1040 IF A=111 THEN B$="_":GOTO 1120
1050 IF A=116 THEN B$="| ":GOTO 1120
1060 IF A=119 THEN B$="¯":GOTO 1120
1070 IF A=76 THEN B$="L":GOTO 1120
1080 IF A=106 THEN B$=" |":GOTO 1120
1090 IF A=32 OR A=96 THEN B$=" ":GO
     TO 1120
1100 IF A=81 THEN B$="●":GOTO 1120
1110 STOP
1120 A$=A$+B$
1130 NEXTI
1140 PRINT#3,A$
1150 NEXTJ:PRINT#6,CHR$(24):CLOSE 6
     ,4
1160 RETURN
1170 REM  PRINT AT
1180 PO=3070+X+40*(Y-2)+81:POKE PO,
     S%(X,Y):RETURN
1190 REM  DISEGNA CONTORNO
1200 PRINT"[CLEAR]"
1210 PRINT" ";:FOR I=1 TO DX:PRINT"
     _";:NEXTI:PRINT" "
1220 FOR I=1 TO DY:PRINTSPC(DX+1);"
     |":NEXTI
1230 RETURN
1240 REM  INIT VARIABILI
1250 DEF FNO(X)=INT(3*RND(1))
1260 DIM A(4,4),P(4,6),S%(40,23),F(
     4)
1270 FOR I=1 TO 4:FOR J=1 TO 4
1280 READ A(I,J):NEXTJ,I
1290 FOR I=1 TO 4:FOR J=1 TO 6
1300 READ P(I,J):NEXTJ,I
1310 FOR J=1 TO DY+2:S%(1,J)=1:S%(D
     X+2,J)=1:NEXTJ
1320 FOR J=1 TO DX+2:S%(J,1)=1:S%(J
     ,DY+2)=1:NEXTJ
1330 FOR I=1 TO 4:READ F(I):NEXTI
1340 RETURN
1350 DATA 116,96,42,116,76,111,116,
     42,42,111,116,76,111,42,96,111
1360 DATA 1,2,4,1,2,4,1,2,3,1,2,3,2
     ,3,4,2,3,4,1,3,4,1,3,4
1370 DATA 116,76,76,111
1380 REM  LISTA
1390 PRINT"[CLEAR]"
1400 DIRECTORY"LAB.*"
1410 GET KEY A$
1420 RETURN
```

ANALISI COMBINATORIA

di Eugenio Coppari

Alcune branche della matematica possono costituire la fonte teorica di inaspettate ricchezze economiche. Ad esempio la scelta di una determinata opzione nel corso di un gioco d'azzardo, dovrebbe essere indotta dalle probabilità che ha di verificarsi un evento successivo da noi preventivato. Sussiste quindi il problema di vagliare quale tra le alternative possibili sia quella che può sortire il migliore effetto finale. Un ragionamento di questo genere è basato soprattutto sul numero di situazioni tra le quali ci troviamo costretti a dover scegliere in determinate fasi del gioco.

Questo articolo ed il conseguente programma si occuperanno del calcolo combinatorio, un settore della matematica che possiede molteplici applicazioni nei più svariati campi della nostra vita quotidiana. Ma passiamo alla trattazione teorica del problema.

Assegnati N oggetti distinti, si dice che essi sono ordinati in un allineamento, quando si collocano in N posti numerati. Per esempio un allineamento dei primi cinque numeri naturali può essere: 5,3,4,1,2.

Viene denominata permutazione di N elementi distinti ogni allineamento degli oggetti stessi. E' quindi intuitivo che due permutazioni sono distinte quando differiscono per il posto occupato da almeno un oggetto. Ad esempio le permutazioni possibili dei primi tre numeri naturali sono: 123, 321, 213, 132, 231, 312. Si può dimostrare, abbastanza agevolmente, che il numero delle permutazioni consentite, di N oggetti diversi, è dato da: 1 per 2 per 3 per 4... per N. Ad esempio se ho a disposizione 4 diversi oggetti il numero di tutti i loro diversi allineamenti è dato da: 1 per 2 per 3 per 4, cioè 16. Collocando al primo posto un oggetto qualunque preso tra i quattro disponibili, avrò la possibilità di scegliere, inizialmente, tra 4 diverse opzioni. Poniamo al secondo posto, uno qualsiasi tra i tre elementi rimasti e otterremo che le scelte attuali sono pari a 4*3. Quindi collochiamo al terzo e quarto posto i due elementi rimasti e otterremo un numero di allineamenti pari a 16. Abbiamo quindi

fornito una dimostrazione dell'asserto nel caso in cui N sia uguale a 4. La dimostrazione della proposizione può essere facilmente estesa a un numero superiore di termini ottenendo la relazione matematica P(N)=1*2*3*4*... N (dove P(N) denota il numero delle permutazioni).

Può sussistere il caso in cui non tutti gli elementi differiscano tra di loro: sarà quindi necessario modificare sostanzialmente l'asserto precedente per poter calcolare il numero delle permutazioni degli oggetti. Il numero delle permutazioni distinte di N oggetti di cui A uguali tra loro, B uguali tra loro e distinti dai precedenti, C uguali tra di loro e distinti dai precedenti, è dato dal prodotto dei primi N numeri naturali diviso per il risultato delle operazioni riportate di seguito: prodotto dei primi A numeri naturali moltiplicati per il prodotto dei primi B numeri naturali moltiplicato per il prodotto dei primi C numeri naturali. Naturalmente A=B=C deve essere uguale al numero naturale N.

In matematica il prodotto dei primi N numeri naturali viene denominato fattoriale di N e si è soliti indicarlo nella seguente maniera: N! Anche nel caso di permutazioni di oggetti non necessariamente distinti esaminiamo alcuni esempi.

Consideriamo 8 oggetti:
q,q,q,g,g,g,g,h di cui tre uguali tra loro e quattro uguali tra loro e distinti dai precedenti. E' evidente che non distinguendo più gli elementi q tra loro le permutazioni degli 8 oggetti non arriveranno a: 1*2*3*4*5*6*7*8 (8!). Distinguiamo con un indice gli oggetti uguali tra loro: q1,q2,q3,q4,g1,g2,g3,h. Consideriamo una qualsiasi delle 8 fattoriali permutazioni (8!): q2,q1,g1,h,g2,g3,q4. Se adesso togliamo l'indice alle tre lettere g, esse possono essere permutate in uno qualsiasi dei 3! modi nella terna dei posti da esse occupati, senza che le permutazioni si distinguano tra loro. In base a ciò gli allineamenti distinti possibili si ridurranno, in questa prima fase a 8 fattoriale diviso tre fattoriale (8!/3!). L'analogo ragionamento applicato alle lettere q porta, naturalmente, al seguente risultato finale P(N)=8!/(3!*4!).

Dopo le permutazioni, passiamo ora ad un altro interessante argomento della analisi combinatoria.

Le disposizioni

Si dice disposizione semplice di N oggetti di classe K ogni allineamento di K oggetti scelti tra gli N. Naturalmente due qualsiasi disposizioni possono differire per gli oggetti contenuti oppure per l'ordine in cui sono disposti. Ad esempio le disposizioni dei primi tre numeri naturali di classe 2 sono: 12,21,13,31,23,32. Il numero delle disposizioni di N oggetti di classe K è data da: N moltiplicato per (N-1) moltiplicato per... (N-K+1).

L'asserto precedente si giustifica nella seguente maniera: si considerino K posti allineati e si scelga un elemento tra gli N da collocare in prima posizione, poi uno qualunque degli (N-1) rimasti da collocare alla seconda locazione e così via sino all'ultimo che verrà collocato in (N-K+1). Un attimo di riflessione dovrebbe essere sufficiente per stabilire la veridicità del teorema precedente. Si dice disposizione con ripetizione di N oggetti, non necessariamente distinti, di classe K ogni allineamento di K oggetti. Ad esempio le disposizioni con ripetizione dei primi 3 numeri naturali di classe 2 sono: 11,12,21,22,13,31,33,23,32. Il numero di questi allineamenti è dato dalla relazione N elevato alla K. La veridicità di questa asserzione può essere verificata dal lettore, prendendo come base la proposizione concernente le disposizioni semplici.

L'ultimo argomento da affrontare è quello delle combinazioni.

Le combinazioni

Si dice combinazione semplice di N elementi di classe K ogni raggruppamento di K oggetti comunque scelti tra gli N (in questo caso non si tiene conto dell'ordine!). Ad esempio le combinazioni dei primi 4 numeri naturali sono le seguenti: 12,13,23,24,14,34. Poichè in

questo caso non si tiene conto dell'ordine, ad esempio: 12 e 21 saranno equivalenti e andranno quindi conteggiati una sola volta. Il numero delle combinazioni di N oggetti di classe K è dato da: N moltiplicato per (N-1) moltiplicato per (N-K+1) diviso K fattoriale.

Per quanto concerne la dimostrazione dell'asserto sarà necessario fare riferimento ad un qualsiasi testo di analisi matematica oppure, avvalendosi delle argomentazioni precedenti, ricavarla autonomamente. Si dice combinazione con ripetizione di N oggetti di classe K ogni raggruppamento di K oggetti comunque scelti tra gli N assegnati, con la convinzione che un elemento può essere ripetuto una o più volte. Due di queste combinazioni sono diverse quando differiscono per almeno un oggetto contenuto o per il numero di volte che esso è contenuto. Ad esempio le combinazioni con ripetizione dei primi 4 numeri naturali di classe 2 sono:
11,12,13,14,22,23,24,33,34,44. Il numero delle combinazioni con ripetizione di N oggetti di classe K è dato da: N moltiplicato per (N+1) moltiplicato per ... (N+K-1) e diviso K fattoriale (K!).

Il programma

Esaurita la parte prettamente teorica passiamo alla descrizione del programma. Il software consente di calcolare permutazioni, disposizioni e combinazioni semplici di N oggetti di classe K. Con K indichiamo, naturalmente, la lunghezza dell'allineamento su cui verranno effettuate le operazioni precedentemente citate. Gli algoritmi utilizzati dal programma prendono spunto dalla precedente trattazione teorica. Dovrebbero quindi risultare sufficientemente autoesplicativi anche per coloro che si avvicinassero per la prima volta a questo genere di trattazione. Nelle righe da 250 sino a 286 troviamo la zona di programma concernente le permutazioni, nelle righe da 306 sino a 366 è collocato invece il calcolo delle disposizioni semplici. Infine da 386 sino a 464 è situato il calcolo delle combinazioni di N oggetti di

```
------*---------------- - - -
VAR.  ! LINEA DEL PROGRAMMA
------*---------------- - - -
      !
A$    !   252   254   256   260   308   310
      !   312   316   322   324   326   330
      !   388   390   392   396   402   404
      !   406   410
C     !   268   272   278
E     !   182   348   426   438
E$    !   178   180   226   228   230
E1$   !   160   504   514   524   534   544
      !   554   590   600   610   620   666
      !   676
E2$   !   160   500   510   520   530   540
      !   550   586   596   606   616   662
      !   672
ED    !   270   272   274
EZ    !   454   478   480
F1$   !   162   504   510   514   524   530
      !   534   544   550   554   580   590
      !   620   630   656   676
F2$   !   162   500   520   540   576   586
      !   616   626   652   672
GD    !   470   472
GT    !   168
H     !   422   426   428   448
H$    !   282   284   362   364   452   454
I     !   424   426   428   430
J     !   342   348   350   358   432   438
      !   440   448
K     !   324   334   336   338   340   344
      !   404   414   416   418   420   424
      !   434
Q     !   346   354   436   444
Q1$   !   158   504   514   520   524   534
      !   540   544   554   580   600   610
      !   630   656   666
Q2$   !   158   500   510   530   550   576
      !   596   606   626   652   662
S$    !   258   260   264   266   270   310
      !   316   390   396
SS    !   476   478   482
T     !   344   346   348   350   352   434
      !   436   438   440   442   502   506
      !   512   516   522   526   532   536
      !   542   546   552   556   578   582
      !   588   592   598   602   608   612
      !   618   622   628   632   654   658
      !   664   668   674   678
W     !   266
W$    !   324   330   404   410
X$    !   338   418
XX$   !   508   518   528   538   548   558
      !   584   594   604   614   624   634
      !   660   670   680
Y     !   310   334   336   338   340   344
      !   346   390   414   416   418   420
      !   434   436
YH$   !   462
Z     !   314   394
ZV    !   364   484
ZZ    !   328   408
```

classe K. Le righe da 500 sino a 682 svolgono una funzione di carattere didattico nei confronti dell'utente che voglia meglio comprendere il calcolo combinatorio: viene visualizzato graficamente il concetto di permutazione di 3 oggetti tra loro distinti, oltre a quello di disposizione e combinazione di 3 elementi di classe 2.

```
100 REM ************************
104 REM * TEMA DEL PROGRAMMA : *
108 REM * CALCOLO COMBINATORIO *
112 REM ************************
116 REM * AUTHOR  SOFTWARE :   *
120 REM * EUGENIO COPPARI      *
122 REM *                      *
124 REM ************************
126 REM *                      *
128 REM *COMPUTER  UTILIZZABILI*
130 REM *                      *
132 REM * COMMODORE VC 20 : NO *
134 REM *                      *
136 REM * COMMODORE 64    : SI *
138 REM *                      *
140 REM * COMMODORE 16    : SI *
142 REM *                      *
144 REM * PLUS 4          : SI *
146 REM *                      *
148 REM * SERIE 4000      : SI *
150 REM *                      *
152 REM * SERIE 8000      : SI *
156 REM ************************
158 Q1$="┌┐[DOWN][2 LEFT]└┘":Q2$="
    [DOWN][2 LEFT]  "
160 E1$="╲╱[DOWN][2 LEFT]╱╲":E2$="
    [DOWN][2 LEFT]  "
162 F1$="┴[DOWN][2 LEFT]┬":F2$="
    [DOWN][2 LEFT]  "
164 POKE 53280,6:POKE 53281,6:PRIN
    T"[CLEAR][GIALLO]"
166 PRINT"[GIALLO][4 DOWN][12 RIGH
    T]╭──────────────╮"
168 FOR GT=1 TO 9:PRINT"[12 RIGHT]
    |              |":NEXTGT
170 PRINT"[12 RIGHT]╰──────────────
    ──╯"
172 PRINT"[HOME][6 DOWN][13 RIGHT]
    1- PERMUTAZIONI"
174 PRINT"[3 DOWN][13 RIGHT]2- DIS
    POSIZIONI"
176 PRINT"[3 DOWN][13 RIGHT]3- COM
    BINAZIONI"
178 GET E$:IF E$="" THEN 178
180 IF ASC(E$)<49 OR ASC(E$)>51 TH
    EN 178
182 IF VAL(MID$(STR$(E),2))<>E THE
    N 178
184 REM ************************
186 REM *                      *
188 REM *  CREAZIONE   DELLA   *
190 REM *                      *
192 REM *  MASCHERINA DEL PRG  *
194 REM *                      *
196 REM ************************
198 PRINT"[CLEAR][3 DOWN][3 RIGHT]
200 PRINT"[5 RIGHT]╭──────────────
    ──────────────╮"
202 PRINT"[5 RIGHT]|
              |"
204 PRINT"[5 RIGHT]|
              |"
206 PRINT"[5 RIGHT]|
              |"
208 PRINT"[5 RIGHT]|
              |"
210 PRINT"[5 RIGHT]|
              |"
212 PRINT"[5 RIGHT]|
              |"
214 PRINT"[5 RIGHT]|
              |"
216 PRINT"[5 RIGHT]|
              |"
218 PRINT"[5 RIGHT]|
              |"
220 PRINT"[5 RIGHT]|
              |"
222 PRINT"[5 RIGHT]|
              |"
224 PRINT"[5 RIGHT]╰──────────────
    ──────────────╯"
226 IF E$="1" THEN 250
228 IF E$="2" THEN 306
230 IF E$="3" THEN 386
232 REM ************************
234 REM *                      *
236 REM *   CALCOLO     DELLE  *
238 REM *                      *
240 REM *     PERMUTAZIONI     *
242 REM *                      *
244 REM * DI N OGGETTI DISTINTI *
246 REM *                      *
248 REM ************************
```

```
250 PRINT"[HOME][6 DOWN][6 RIGHT]N
    UMERO DEGLI ELEMENTI :";
252 GET A$:IF A$="" THEN 252
254 IF ASC(A$)=13 THEN 264
256 IF ASC(A$)<48 OR ASC(A$)>57 TH
    EN 252
258 IF LEN(S$)=2 THEN 252
260 PRINTA$;:S$=S$+A$
262 GOTO 252
264 IF VAL(S$)>32 THEN PRINT"[HOME
    ][9 DOWN][7 RIGHT]NON E' POSSI
    BILE INSERIRE "
266 IF VAL(S$)>32 THEN PRINT"[DOWN
    ][7 RIGHT]  PIU' DI 32 ELEMEN
    TI":FOR W=1 TO 3000:NEXT: RUN
268 C=1
270 FOR ED=1 TO VAL(S$)
272 C=C*ED
274 NEXTED
276 PRINT"[HOME][9 DOWN][6 RIGHT]N
    UMERO DELLE PERMUTAZIONI :"
278 PRINT"[DOWN][12 RIGHT]";C
280 PRINT"[DOWN][6 RIGHT]ESEMPIO D
    IDATTICO ?"
282 GET H$:IF H$="" THEN 282
284 IF H$="S" THEN 466
286  RUN
288 REM *************************
290 REM *                       *
292 REM *   CALCOLO      DELLE  *
294 REM *                       *
296 REM *     DISPOSIZIONI      *
298 REM *                       *
300 REM * DI N OGGETTI DISTINTI *
302 REM *                       *
304 REM *************************
306 PRINT"[HOME][6 DOWN][6 RIGHT]N
    UMERO ELEMENTI: ";
308 GET A$:IF A$="" THEN 308
310 IF ASC(A$)=13 THEN Y=VAL(S$):G
    OTO 320
312 IF ASC(A$)<48 OR ASC(A$)>57 TH
    EN 308
314 Z=Z+1:IF Z>10 THEN 308
316 PRINTA$;:S$=S$+A$
318 GOTO 308
320 PRINT"[HOME][9 DOWN][6 RIGHT]G
    RUPPO K :";
322 GET A$:IF A$="" THEN 322
324 IF ASC(A$)=13 THEN K=VAL(W$):G
    OTO 334
326 IF ASC(A$)<48 OR ASC(A$)>57 TH
    EN 322
328 ZZ=ZZ+1:IF ZZ>16 THEN 322
330 PRINTA$;:W$=W$+A$
332 GOTO 322
334 IF K>Y THEN PRINT:PRINT"[2 DOW
    N][7 RIGHT]NON POSSONO ESISTER
    E TALI"
336 IF K>Y THEN PRINT TAB(47);"DIS
    POSIZIONI,POICHE'  K>Y"
338 IF K>Y THEN GET X$:IF X$="" TH
    EN 338
340 IF K>Y THEN  RUN
342 J=1
344 T=Y-K+1
346 FOR Q=T TO Y
348 IF 1.70141183E+38/T<J THEN 458
350 J=T*J
352 T=T+1
354 NEXTQ
356 PRINT"[HOME][11 DOWN][6 RIGHT]
    NUMERO DELLE DISPOSIZIONI :"
358 PRINT"[DOWN][12 RIGHT]";J
360 PRINT"[DOWN][6 RIGHT]ESEMPIO D
    IDATTICO ?"
362 GET H$:IF H$="" THEN 362
364 IF H$="S" THEN ZV=1:GOTO 466
366  RUN
368 REM *************************
370 REM *                       *
372 REM *   CALCOLO      DELLE  *
374 REM *                       *
376 REM *     COMBINAZIONI      *
378 REM *                       *
380 REM * DI N OGGETTI DISTINTI *
382 REM *                       *
384 REM *************************
386 PRINT"[HOME][6 DOWN][6 RIGHT]N
    UMERO ELEMENTI: ";
388 GET A$:IF A$="" THEN 388
390 IF ASC(A$)=13 THEN Y=VAL(S$):G
    OTO 400
392 IF ASC(A$)<48 OR ASC(A$)>57 TH
    EN 388
394 Z=Z+1:IF Z>10 THEN 308
396 PRINTA$;:S$=S$+A$
398 GOTO 388
400 PRINT"[HOME][9 DOWN][6 RIGHT]G
    RUPPO K :";
402 GET A$:IF A$="" THEN 402
404 IF ASC(A$)=13 THEN K=VAL(W$):G
    OTO 414
406 IF ASC(A$)<48 OR ASC(A$)>57 TH
```

```
   EN 402
408 ZZ=ZZ+1:IF ZZ>16 THEN 402
410 PRINTA$;:W$=W$+A$
412 GOTO 402
414 IF K>Y THEN PRINT:PRINT"[2 DOW
    N][7 RIGHT]NON POSSONO ESISTER
    E TALI"
416 IF K>Y THEN PRINT TAB(47);"COM
    BINAZIONI,POICHE'  K>Y"
418 IF K>Y THEN GET X$:IF X$="" TH
    EN 338
420 IF K>Y THEN  RUN
422 H=1
424 FOR I=1 TO K
426 IF 1.70141183E+38/I<H THEN 458
428 H=H*I
430 NEXTI
432 J=1
434 T=Y-K+1
436 FOR Q=T TO Y
438 IF 1.70141183E+38/T<J THEN 458
440 J=T*J
442 T=T+1
444 NEXTQ
446 PRINT"[HOME][11 DOWN][6 RIGHT]
    NUMERO DELLE COMBINAZIONI :"
448 PRINT"[DOWN][12 RIGHT]";J/H
450 PRINT"[DOWN][6 RIGHT]ESEMPIO D
    IDATTICO ?"
452 GET H$:IF H$="" THEN 452
454 IF H$="S" THEN EZ=1:GOTO 466
456  RUN
458 PRINT:PRINT"[2 DOWN][7 RIGHT]I
    L CALCOLO ECCEDE I LIMITI"
460 PRINT TAB(48);"CONSENTITI DAL
    COMPUTER"
462 GET YH$:IF YH$="" THEN 462
464  RUN
466 PRINT"[CLEAR]";
468 PRINT"[19 RIGHT]╭───────────╮"
470 FOR GD=1 TO 2
472 PRINT"[19 RIGHT]|           |":
    NEXTGD
474 PRINT"[19 RIGHT]╰───────────╯"
476 SS=SS+1
478 IF EZ=1 AND SS=3 THEN 652
480 IF EZ=1 THEN PRINT"[3 DOWN]"
482 IF SS<6 THEN 468
484 IF ZV=1 THEN 574
486 REM ************************
488 REM *                      *
490 REM * DIDATTICA   CON  LE  *
492 REM *                      *
494 REM *    PERMUTAZIONI      *
496 REM *                      *
498 REM ************************
500 PRINT"[HOME][DOWN] PERMUTAZION
    E 1 :[4 RIGHT]";Q2$;"[UP][RIGH
    T]";E2$;"[UP][RIGHT]"F2$
502 FOR T=1 TO 200:NEXTT
504 PRINT"[HOME][DOWN] PERMUTAZION
    E 1 :[4 RIGHT]";Q1$;"[UP][RIGH
    T]";E1$;"[UP][RIGHT]"F1$
506 FOR T=1 TO 150:NEXTT
508 GET XX$:IF XX$="" THEN 500
510 PRINT"[HOME][5 DOWN] PERMUTAZI
    ONE 2 :[4 RIGHT]";E2$;"[UP][RI
    GHT]";Q2$;"[UP][RIGHT]";F1$
512 FOR T=1 TO 200:NEXTT
514 PRINT"[HOME][5 DOWN] PERMUTAZI
    ONE 2 :[4 RIGHT]";E1$;"[UP][RI
    GHT]";Q1$;"[UP][RIGHT]"F1$
516 FOR T=1 TO 150:NEXTT
518 GET XX$:IF XX$="" THEN 510
520 PRINT"[HOME][9 DOWN] PERMUTAZI
    ONE 3 :[4 RIGHT]";F2$;"[UP][RI
    GHT]";Q1$;"[UP][RIGHT]"E2$
522 FOR T=1 TO 200:NEXTT
524 PRINT"[HOME][9 DOWN] PERMUTAZI
    ONE 3 :[4 RIGHT]";F1$;"[UP][RI
    GHT]";Q1$;"[UP][RIGHT]"E1$
526 FOR T=1 TO 150:NEXTT
528 GET XX$:IF XX$="" THEN 520
530 PRINT"[HOME][13 DOWN] PERMUTAZ
    IONE 4 :[4 RIGHT]";F1$;"[UP][R
    IGHT]";E2$;"[UP][RIGHT]"Q2$
532 FOR T=1 TO 200:NEXTT
534 PRINT"[HOME][13 DOWN] PERMUTAZ
    IONE 4 :[4 RIGHT]";F1$;"[UP][R
    IGHT]";E1$;"[UP][RIGHT]"Q1$
536 FOR T=1 TO 150:NEXTT
538 GET XX$:IF XX$="" THEN 530
540 PRINT"[HOME][17 DOWN] PERMUTAZ
    IONE 5 :[4 RIGHT]";E2$;"[UP][R
    IGHT]";F2$;"[UP][RIGHT]"Q1$
542 FOR T=1 TO 200:NEXTT
544 PRINT"[HOME][17 DOWN] PERMUTAZ
    IONE 5 :[4 RIGHT]";E1$;"[UP][R
    IGHT]";F1$;"[UP][RIGHT]"Q1$
546 FOR T=1 TO 150:NEXTT
548 GET XX$:IF XX$="" THEN 540
550 PRINT"[HOME][21 DOWN] PERMUTAZ
    IONE 6 :[4 RIGHT]";Q2$;"[UP][R
    IGHT]";F1$;"[UP][RIGHT]"E2$
```

```
552 FOR T=1 TO 200:NEXTT
554 PRINT"[HOME][21 DOWN] PERMUTAZ
    IONE 6 :[4 RIGHT]";Q1$;"[UP][R
    IGHT]";F1$;"[UP][RIGHT]"E1$
556 FOR T=1 TO 150:NEXTT
558 GET XX$:IF XX$="" THEN 550
560  RUN
562 REM ************************
564 REM *                      *
566 REM *  DIDATTICA  CON  LE  *
568 REM *                      *
570 REM *     DISPOSIZIONI     *
572 REM *                      *
574 REM ************************
576 PRINT"[HOME][DOWN] DISPOSIZION
    E 1 :[5 RIGHT]";F2$;"[UP][3 RI
    GHT]";Q2$
578 FOR T=1 TO 200:NEXTT
580 PRINT"[HOME][DOWN] DISPOSIZION
    E 1 :[5 RIGHT]";F1$;"[UP][3 RI
    GHT]";Q1$
582 FOR T=1 TO 150:NEXTT
584 GET XX$:IF XX$="" THEN 576
586 PRINT"[HOME][5 DOWN] DISPOSIZI
    ONE 2 :[5 RIGHT]";E2$;"[UP][3
    RIGHT]";F2$
588 FOR T=1 TO 200:NEXTT
590 PRINT"[HOME][5 DOWN] DISPOSIZI
    ONE 2 :[5 RIGHT]";E1$;"[UP][3
    RIGHT]";F1$
592 FOR T=1 TO 150:NEXTT
594 GET XX$:IF XX$="" THEN 586
596 PRINT"[HOME][9 DOWN] DISPOSIZI
    ONE 3 :[5 RIGHT]";Q2$;"[UP][3
    RIGHT]";E2$
598 FOR T=1 TO 200:NEXTT
600 PRINT"[HOME][9 DOWN] DISPOSIZI
    ONE 3 :[5 RIGHT]";Q1$;"[UP][3
    RIGHT]";E1$
602 FOR T=1 TO 150:NEXTT
604 GET XX$:IF XX$="" THEN 596
606 PRINT"[HOME][13 DOWN] DISPOSIZ
    IONE 4 :[5 RIGHT]";E2$;"[UP][3
     RIGHT]";Q2$
608 FOR T=1 TO 200:NEXTT
610 PRINT"[HOME][13 DOWN] DISPOSIZ
    IONE 4 :[5 RIGHT]";E1$;"[UP][3
     RIGHT]";Q1$
612 FOR T=1 TO 150:NEXTT
614 GET XX$:IF XX$="" THEN 606
616 PRINT"[HOME][17 DOWN] DISPOSIZ
    IONE 5 :[5 RIGHT]";F2$;"[UP][3
     RIGHT]";E2$
618 FOR T=1 TO 200:NEXTT
620 PRINT"[HOME][17 DOWN] DISPOSIZ
    IONE 5 :[5 RIGHT]";F1$;"[UP][3
     RIGHT]";E1$
622 FOR T=1 TO 150:NEXTT
624 GET XX$:IF XX$="" THEN 616
626 PRINT"[HOME][21 DOWN] DISPOSIZ
    IONE 6 :[5 RIGHT]";Q2$;"[UP][3
     RIGHT]";F2$
628 FOR T=1 TO 200:NEXTT
630 PRINT"[HOME][21 DOWN] DISPOSIZ
    IONE 6 :[5 RIGHT]";Q1$;"[UP][3
     RIGHT]";F1$
632 FOR T=1 TO 150:NEXTT
634 GET XX$:IF XX$="" THEN 626
636  RUN
638 REM ************************
640 REM *                      *
642 REM *  DIDATTICA  CON LE   *
644 REM *                      *
646 REM *     COMBINAZIONI     *
648 REM *                      *
650 REM ************************
652 PRINT"[HOME][DOWN] COMBINAZION
    E 1 :[5 RIGHT]";F2$;"[UP][3 RI
    GHT]";Q2$
654 FOR T=1 TO 200:NEXTT
656 PRINT"[HOME][DOWN] COMBINAZION
    E 1 :[5 RIGHT]";F1$;"[UP][3 RI
    GHT]";Q1$
658 FOR T=1 TO 150:NEXTT
660 GET XX$:IF XX$="" THEN 652
662 PRINT"[HOME][9 DOWN] COMBINAZI
    ONE 2 :[5 RIGHT]";Q2$;"[UP][3
    RIGHT]";E2$
664 FOR T=1 TO 200:NEXTT
666 PRINT"[HOME][9 DOWN] COMBINAZI
    ONE 2 :[5 RIGHT]";Q1$;"[UP][3
    RIGHT]";E1$
668 FOR T=1 TO 150:NEXTT
670 GET XX$:IF XX$="" THEN 662
672 PRINT"[HOME][17 DOWN] COMBINAZ
    IONE 3 :[5 RIGHT]";F2$;"[UP][3
     RIGHT]";E2$
674 FOR T=1 TO 200:NEXTT
676 PRINT"[HOME][17 DOWN] COMBINAZ
    IONE 3 :[5 RIGHT]";F1$;"[UP][3
     RIGHT]";E1$
678 FOR T=1 TO 150:NEXTT
680 GET XX$:IF XX$="" THEN 672
682  RUN
```

Errata Addendum Meteo casa

Non era un pesce d'Aprile! Per un errore di impaginazione sul numero scorso, nell'articolo riguardante "*Meteo casa*" è stata involontariamente omessa l'ultima parte del listato. Eccola quindi qui di seguito:

```
2900 IF X%<=24 THEN PRINT TAB(32
)"[GIALLO]ORE";X%
2910 IF X%>24 THEN X%=X%-24:PRIN
T TAB(32)"[GIALLO]ORE";X%
2920 FOR K=1 TO 8000:NEXT
2930 FOR Y=0 TO 1:POKE (C+25+40*
Y),A:NEXT
2940 FOR Y=2 TO 6:POKE (C+25+40*
Y),E:NEXT
2950 FOR Y=7 TO 15:POKE (C+25+40*
Y),O:NEXT
2960 FOR Y=16 TO 22:POKE (C+25+40*
Y),D:NEXT
2970 FOR X=26 TO 27:FOR Y=0 TO
1:POKE (C+X+40*Y),A:NEXT Y
2980 FOR Y=2 TO 7:POKE (C+X+40*Y
),U:NEXTY
2990 POKE (C+X+40*8),B
3000 FOR Y=9 TO 15:POKE (C+X+40*
Y),O:NEXT Y
3010 FOR Y=16 TO 22:POKE (C+X+40
*Y),D:NEXT Y:NEXT X
3020 POKE (C+26+40*2),E
3030 FOR X=28 TO 31:FOR Y=0 TO
1:POKE (C+X+40*Y),A:NEXT Y
3040 FOR Y=2 TO 7:POKE (C+X+40*Y
),U:NEXT Y
3050 FOR Y=8 TO 10:POKE (C+X+40*
Y),B:NEXT Y
3060 FOR Y=11 TO 15:POKE (C+X+40
*Y),O:NEXT Y
3070 FOR Y=16 TO 22:POKE (C+X+40
*Y),D:NEXT Y:NEXT X
3080 POKE (C+30+40*11),B:FOR Y=11
 TO 12:POKE (C+31+40*Y),B:NEX
T Y
3090 X%=X%+8:PRINT"[HOME]":PRINT TA
B(32)"       ":PRINT"[HOME]"
3100 IF X%<=24 THEN PRINT TAB(32
)"ORE";X%
3110 IF X%>24 THEN X%=X%-24:PRINT
 TAB(32)"ORE";X%
3120 FOR K=1 TO 8000:NEXT
3130 FOR X=32 TO 38:FOR Y=2 TO
7:POKE (C+X+40*Y),U:NEXT Y
3140 FOR Y=8 TO 14:POKE (C+X+40*
Y),B:NEXT Y
3150 FOR Y=15 TO 22:POKE (C+X+40
*Y),D:NEXT Y:NEXT X
3160 FOR Y=13 TO 14:POKE (C+32+4
0*Y),O:NEXT Y
3170 FOR X=35 TO 38:POKE (C+X+40
*15),B:NEXT X
3180 FOR X=36 TO 38:POKE (C+X+40
*16),B:NEXT X
3190 X%=X%+8:PRINT"[HOME]":PRINT TA
B(32)"       ":PRINT"[HOME]"
3200 IF X%<=24 THEN PRINT TAB(32
)"ORE";X%
3210 IF X%>24 THEN X%=X%-24 :PRI
NT TAB(32)"ORE";X%
3220 FOR K=1 TO 8000:NEXT
3230 PRINT"[CLEAR][NERO]":POKE 532
81,5
3300 PRINT TAB(8)"[4 DOWN]PREMI F1
PER UN'ALTRA PREVISIONE"
3310 PRINT TAB(8)"[4 DOWN]PREMI F3
PER USCIRE"
3320 GET B$
3330 IF B$=CHR$(133) THEN GOTO
100
3340 IF B$=CHR$(134) THEN PRINT"
[CLEAR][CELESTE]":POKE 53280,
14:POKE 53281,6:NEW
3350 IF B$<>CHR$(133) OR B$<>CHR
$(134) THEN GOTO 3320
```

Non ci sono dubbi la Hannover Messe '85 è la manifestazione fieristica più importante che esiste in Europa.

Situata alla periferia della città non ha avuto problemi, sia di estensioni che di organizzazioni logistiche di ogni genere.

Per dare un'idea coerente si può affermare che tutti i padiglioni che ospitano il settore informatico, che costituiscono il cosiddetto CeBit, si estendono per una superficie pari a quella di tutta la Fiera Internazionale di Milano.

Guida dell'Hannover Messe '85 alla mano, la delegazione della Systems Editoriale si reca, quale primo e più importante appuntamento, allo stand della Commodore.

Su un area di tutto rispetto la Commodore Büromaschinen Gbh ha sfoggiato un vasto numero di novità interessanti. Dai nuovi home fino ad arrivare nientepopodimeno che ad un sistema mini con un massimo di terminali tutti facenti parte di nuovi prodotti di futuro successo.

Proseguendo, diremo in verità, con passo veloce abbiamo visitato molti altri stand soffermandoci laddove venivano presentati prodotti sia hardware che software o collegabili ai Commodore.

Nelle prossime pagine troviamo le foto e alcune brevi descrizioni inerenti ai prodotti della Commodore. Sarà poi nostra premura recensire sia il software che l'hardware di altre case che comunque sono inerenti ai prodotti in particolare della fascia home e personal.

Maggiori dettagli sul C 128, il PC 10 nonchè sull'LCD saranno ragione di articoli futuri che naturalmente saranno corredati di listati e prove.

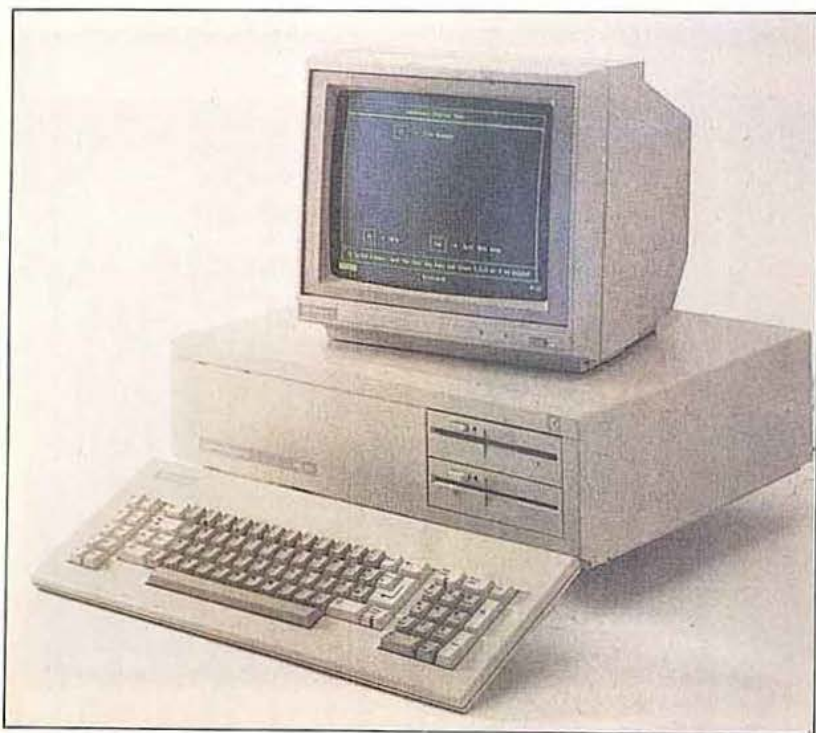
COMMODORE 8296 D

L'ultima versione della serie CBM 8000 si presenta finalmente compatta e veramente completa. La Commodore con l'8296D ha voluto certo soddisfare alle richieste dell'utenza che voleva un sistema che non richiedesse tre blocchi a se stanti collegati elettricamente e logicamente: Unità centrale, stampante e floppy Drive.

Come si può osservare dalla fotografia, nella parte frontale trovano posto due drive per floppy da 5" 1/4 la cui formattazione globale raggiunge: 2 MBytes.

L'evoluzione dal primo 8296 ci porta ad avere due sistemi operativi, il primo disponibile all'accensione: con il BASIC 4.0; il secondo chiamato LOS96 che abilita qualche nuovo comando utile per la programmazione.

Il Kit LOS96 permette di vedere 64 KBytes per i normali lavori, mentre per applicazioni il Linguaggio macchina la memoria disponibile si estende fino a 118 KBytes.

Nonostante la novità del prodotto riteniamo che l'8296D sarà la versione che chiede la serie 8000 in quanto mantiene anche in questa veste la completa e positiva compatibilità con i suoi processori; fatto questo molto importante per il notevole software professionale disponibile.

PC 10 e PC 20

Non poteva certo mancare all'appello della compatibilità PC anche la Commodore. Infatti dopo avere avuto notizie "da corridoio" ecco apparire sul mercato il PC 10.

Quali sono le caratteristiche hardware Microprocessore 8088 funzionante a 4.77 Mhz.

Memoria: 256 KBytes espansibili fino ad un massimo di 640 KBytes8 KBytes ROM (che include il BIOS)

Display: monitor monocromatico a 12 pollici

Memoria di Massa: modello PC 10: due drive tipo slim 5 1/4 di pollice con capacità di 360 KBytes formato Ibm modello PC 20: 1 drive slim 5" 1/4 360 KBytes + hard disc da 10 MBytes

PC 20

La versione PC 20 vede solamente la disponibilità di un solo drive a 360 K, ma al suo interno trova posto un sistema a Winchester da 10 MBytes.

PC 10

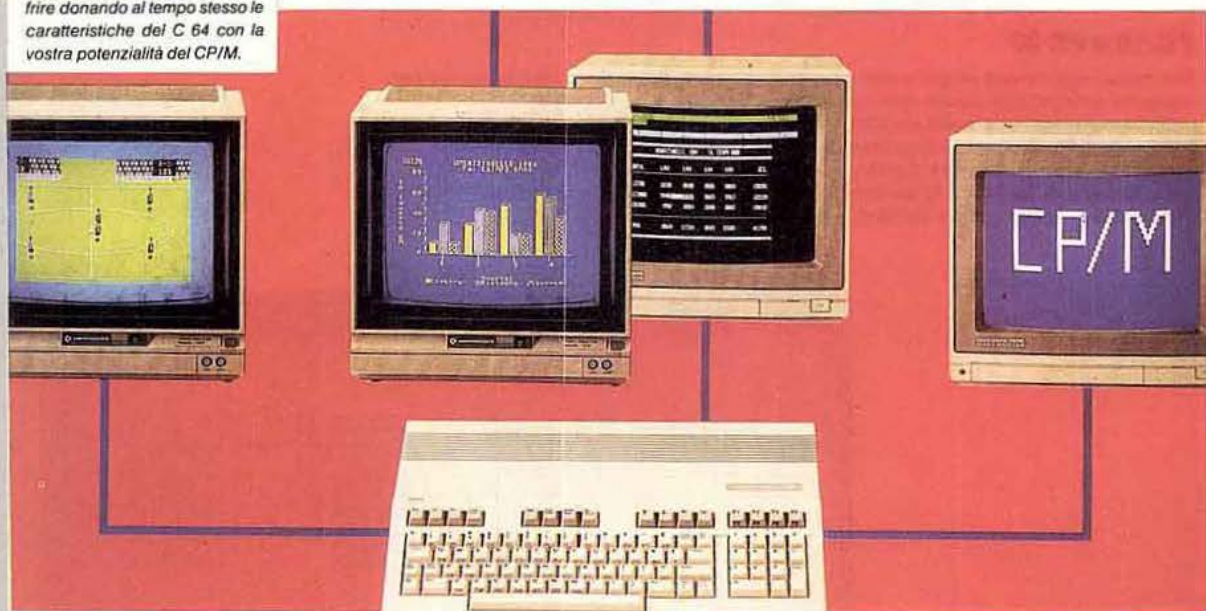
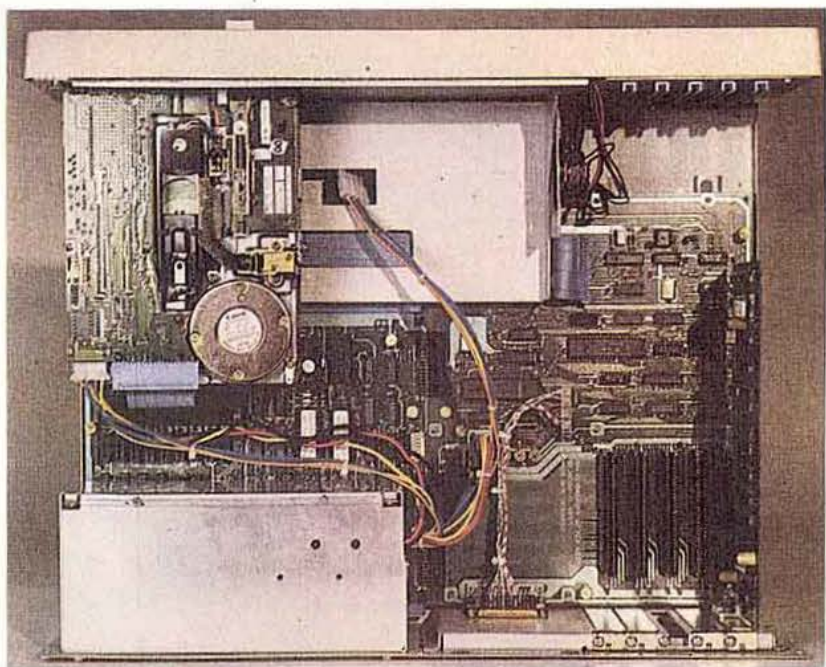
La vera compatibilità fra sistemi non è solo limitata al software, ma anche all'hardware.

Si può verificare, pur solo con una occhiata alle foto, che anche l'hardware risponde alle necessità di intercambiabilità. Infatti possiamo osservare la disposizione della zona espansioni perfettamente compatibile IBM PC.

C 128

Molti fedeli di Commodore aspettavano certo con ansia un nuovo modello di Home Computer che rasentasse il livello di vero e proprio personal. E' il caso di C 128, un nuovo nato di sicuro successo.

Vuoi il C 64? Accendi il C 128 ed ecco il tuo 64. Vuoi il sistema operativo CP/M? Ed ecco con un semplice comando il CP/M. Vuoi 80 colonne? Premi un tasto ed hai 80 colonne.

E' forse il sistema più completo che la Commodore possa offrire donando al tempo stesso le caratteristiche del C 64 con la vostra potenzialità del CP/M.

Tutto il software disponibile sul C 64 può "girare" anche sul C 128. In particolar modo è stato mantenuto nella progettazione hardware anche quel particolare circuito integrato chiamato SID responsabile degli ottimi effetti sonori che già conosciamo. Il microprocessore è sempre il 6510.

Operativamente possiamo attivare il BASIC versione 7.0 che ci permette, tramite il microprecessore 8502, di gestire 128 K RAM, espansibile fino a 512 K. con una serie di 140 comandi.

La versione CP/M disponibile è quella denominata 3.0 ed anche in questa performance il C 128 ha disponibili 128 K RAM espansibili come nel precedente caso. Il CP/M viene automaticamente attivato il microprocessore Z80 A.

C 116

Sembra un mini PLUS 4 vero? No! E' semplicemente la versione compatta del C 16. Largo una spanna, alto due dita potrebbe quasi essere un tascabile.

Anche la Commodore, in questa versione di Home Computer, ha voluto adottare i tastini di ... gqmma.

Arriverà in Italia?

Serie 900

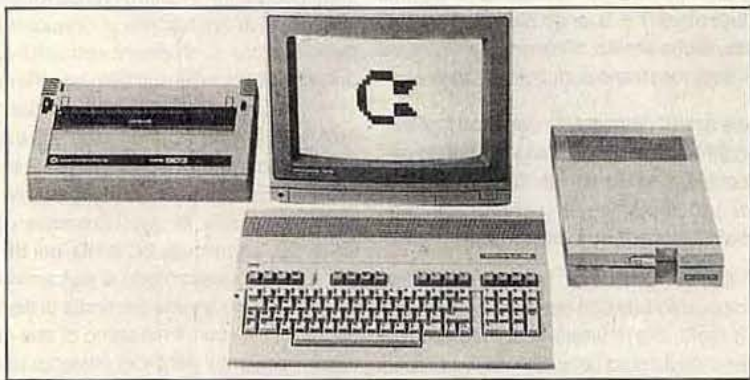
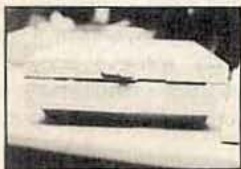
Non certo un Home Computer. Nemmeno un personal. Tantomeno un professional.

I computer della serie 900 non rientrano in quelle categorie, ma si possono collocare nella fascia dei Mini computers.

Laddove può aderire una necessità di multi terminali (massimo 4), laddove si vuole un sistema operativo evoluto (UNIX), laddove si voglia una grafica da "sballo" ecco ... quello è il posto di un serie 900.

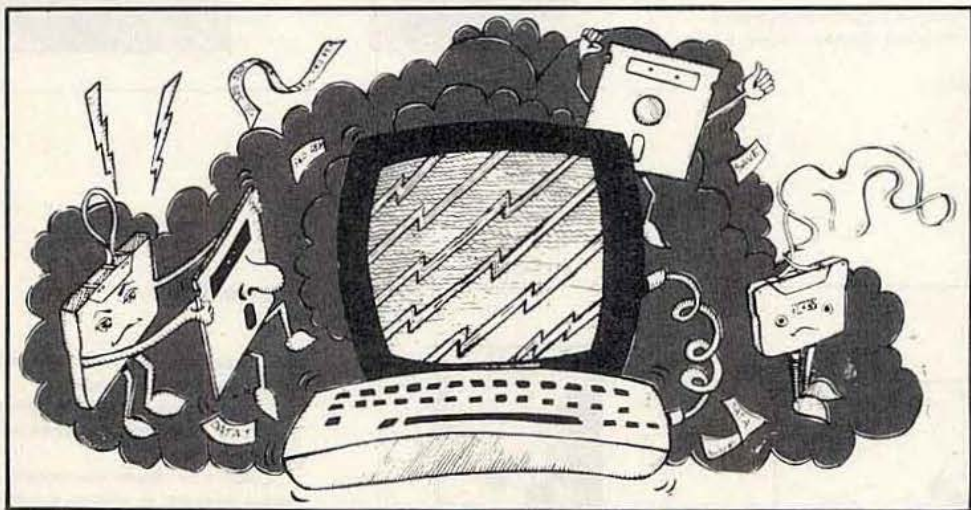
Con questo sistema la Commodore ha volutc varcare la soglia dei mini e poi forse più su ancora. C'è solo da domandarsi: sarà un passo giusto?

Drive 1571

Il nuovo drive 1571 fornisce finalmente una velocità di trasferimento dati cinque volte superiori rispetto al 1541. Fatto triste però e che il C 128 in evoluzione C 64 con il drive 1571 mantiene la vecchia "lenta" velocità.

C 128

Come possiamo notare dalla fotografia, possiamo con il C 128 utilizzare le periferiche del C 64: dalla stampante MPS 803 al registratore 1530, dalla stampante-plotter 1520 al Drive 1541, ect. L'interfacciamento infatti prevede: BS 232 sempre livello TTL, User Port (8 bit paralleli), bus seriale, una porta per registrare, due porte joystick, uscite audio Video e radiofrequenze, porta per cartridge, ect.

IL PROGRAMMA ARCHIVIO

di **Tullio Spezia**

Con questo articolo si viene incontro a quella parte dei nostri lettori che possiede il drive VC-1541 della Commodore per il C-64, e dispone di un certo numero di programmi (magari alcune centinaia) su molti floppy. Come trovare un programma 'salvato' chissà dove fra tutti quei dischi? E quando bisogna fare presto, molto presto, altrimenti l'amico a cui si vuol mostrarlo comincia ad annoiarsi?

Aveveamo cominciato con tanti foglietti, scritti bene in stampatello poi presto pasticciati. E le identity dei floppy? Ci vuole un po' d'ordine, un sistema, un programma che faccia da archivio.

Cominciamo con le ID, le identity. Sappiamo tutti che ogni disco può avere un titolo, ma ricordate che il floppy drive tiene conto solo delle ID e che se ci sono dei doppioni il VC-1541 può andare in confusione. Con le forbici, o meglio con l'apposita macchinetta, facciamo la finestrella sul bordo del disco per potere usare entrambe le facce, diminuendo ingombro e costi di formazione del nostro archivio. Ora diamo un po' di ordine al tutto. Si possono usare, per le ID, sia lettere che numeri, anche misti. Se usiamo solo numeri, chiamando 00 la facciata anteriore del primo disco e 01 la facciata posteriore, è chiaro che con questo sistema si arriva appena al cinquantesimo disco che si chiamerà con la ID=98 anteriormente e 99 sull'altro lato. Perciò è meglio usare le lettere, cominciando con AA e AB per il primo disco, AC e AD per il secondo, poi AE ed AF per il successivo. Esaurito l'intero alfabeto come seconda lettera, si può riprendere con BA e BB, seguito da BC e BD, poi BE e BF, ecc. In questo modo si può arrivare ad identificare alcune centinaia di dischi usando la ID con il massimo di due caratteri consentiti dal DOS (sistema operativo del disco).

L'obiettivo

Eccoci al punto: dobbiamo creare il programma per cercare i programmi, per ritrovare quel certo programma su quel certo disco! Perchè la cosa sia gradevole all'uso, dovrà essere:

1/ veloce, anzi velocissima;

2/ facile, anzi facilissima;

3/ semplice, anzi semplicissima per le aggiunte e le correzioni.

Un programma archivio normalmente sfrutta un Data Base, sia pure del tipo più elementare. Ma ogni medaglia ha il suo rovescio, e per un data base necessita un manuale da leggere e tenere sotto mano. Spesso è necessario un disco a parte, da formattare e tenere a disposizione esclusiva. Inoltre qualunque data base richiede un tempo non indifferente per il load del programma stesso e per il successivo caricamento del file sequenziale che si tira dietro. Inoltre ancora inizializzato, occorre rispondere a tante domande... Noi abbiamo bisogno di qualcosa che si occupi strettamente della ricerca di un titolo di un file, tralascian-

do tutte le complesse altre possibilità tipiche dei data base. Per esempio, come si scrive il titolo di quel gioco che stiamo cercando: ZAXON, ZAKSON, ZAXOM, ZASSON, ZHASSON, ZAXXON.... Con un data base, ogni volta che si sbaglia, il programma si rifiuta di rispondere, pretende da noi l'esattezza scrupolosa, non ci viene incontro e non ci fa sapere su quale disco è stato copiato. Per avviare l'inconveniente deve bastare la Z iniziale, oppure ZA ed il nostro programma, se non ci ricordiamo meglio, ci indicherà tutti i programmi che cominciano con Z (oppure con ZA, oppure con ZAX, perchè è sufficiente la parte iniziale del nome)

Solo così il programma sarà facile, anzi facilissimo da usare. Non avevamo alcuna perdita di tempo dovuta alla nostra imprecisione, ed otterremo una sola risposta (la sola che serve), la ID del disco...

Useremo i DATA e ... niente BASE...

Effettivamente il programma-cerca-programmi qui presentato fa uso dei DATA, come potrete vedere facendolo funzionare e poi andando a guardare in fondo al listato. Inoltre per fare delle variazioni al nostro elenco, come aggiunte e cancellazioni, basta chiedere il LIST ed operare direttamente la modifica, come si usa normalmente in Basic (si cancella una linea semplicemente battendo il suo numero seguito dal Return, e si aggiunge un'altra linea dandole un numero intermedio fra quelli delle linee tra cui vuole inserirla).

Per compilare l'elenco e cercarvi dentro si usano i tasti funzione, i cosiddetti tasti F1 F3 eccetera, che saranno sempre bene in vista sul video, senza bisogno di doversi rileggere la lezione, con delle scritte che ci guidino in ogni momento. L'uso dei tasti funzione ci darà l'accesso immediato a ciò che vogliamo, senza inutili giri viziosi attraverso i soliti menu e relativi rimandi.

Possiamo sempre dare un'occhiata al nostro elenco facendo il LIST, come pure possiamo stamparlo su carta come un qualsiasi listato. Se cominciamo ordinatamente ad immettere i titoli dal primo disco (il disco AA/AB) il tutto avrà anche un bell'aspetto d'insieme, anche se non è proprio necessario dal punto di vista della ricerca automatica.

I vantaggi

Per alcune centinaia di titoli bastano poche decine di blocchi nel disco; per il programma vero e proprio ne bastano sette e ciò è positivo per diversi motivi. In primo luogo batte rapidamente il programma stesso, poi può essere tenuto sullo stesso disco su cui tenete le vostre routines favorite come i due o tre copy soliti, inoltre non vi porta via un disco, ma solo pochi blocchi di un disco. Infine ogni volta che lo modificate (aggiungendo nuovi titoli all'archivio), fate il SAVE WITH REPLACE e cioè sostituite sul disco il nuovo elenco al precedente battendo:

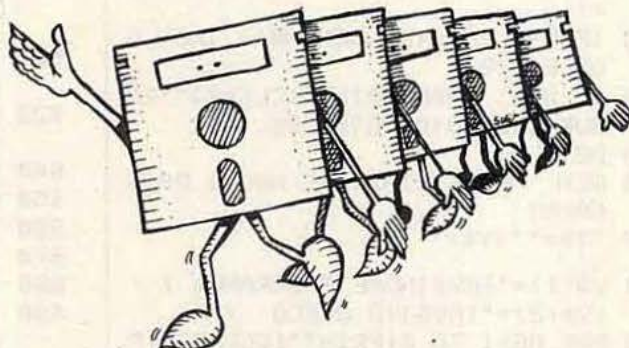
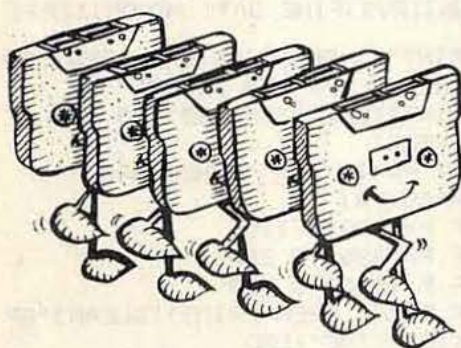
SAVE" 0:ELENCOPROGRAMMI",8

anzichè

SAVE"ELENCO PROGRAMMI",8.

Si risparmia tempo e spazio sul disco; anche di questo troverete l'annotazione pro-memoria che salterà fuori al punto giusto del programma.

Il programma è tutto in Basic ed è possibile modificarlo a proprio gradimento per quei particolari che si vogliono personalizzare; anzi, è proprio l'occasione che andavate cercando di avere una utility di uso continuo, di vostro gusto e da rendere facilmente tutta vostrT, magari adattandola ad altri usi.

Ecco il listato: buon lavoro, e soprattutto buon divertimento. Vi sorprenderà.

```
100 REM  ELENCO PROGRAMMI  (BY I2T
    ZS)
110 POKE 53280,0:POKE 53281,0:PRIN
    T"[GIALLO]"
120 CLR :PRINT"[CLEAR]":MM=631
130 PRINT"[5 DOWN]"SPC(12)"ELENCO
    PROGRAMMI"
140 PRINT"[DOWN]"SPC(10)"E RICERCA
     ALFABETICA"
150 PRINT"[12 DOWN]F1=RICERCA
    F3=SCRITTURA        F7=FINE"
160 GET S$:IF S$="" THEN 160
170 S=ASC(S$)
180 IF S=136 THEN PRINT"[CLEAR]"SP
    C(18)"FINE":END
190 IF S=133 THEN 220
200 IF S=134 THEN 350
210 RETURN
220 REM  RICERCA PROGRAMMI NELLA M
    EMORIA
230 X$="":PRINT"[CLEAR]"SPC(6)"[RV
    S]RICERCA PROGRAMMI NELLA MEMO
    RIA"
240 PRINT:PRINT:PRINT
250 RESTORE :READ NN
260 PRINT"[2 DOWN]"SPC(7)"(SI PUO'
     BATTERE ANCHE SOLO"
270 PRINT"[2 DOWN]"SPC(7)"LA PARTE
     INIZIALE DEL NOME)"
280 PRINT"[7 DOWN]"SPC(10)"[RVS]NO
    ME DEL PROGRAMMA  ?":PRINT"[DO
    WN]"SPC(5):INPUT N$
290 IF ASC(N$)=136 THEN  RUN
300 PRINT"[CLEAR]"
310 READ L:READ X$:IF L=NN THEN BB
    =1
320 IF N$=LEFT$(X$,LEN(N$)) THEN G
    OSUB 570
330 IF BB=1 THEN PRINT"[CLEAR]":GO
    SUB 610:BB=0:GOTO 220
340 GOTO 310
350 REM  MEMORIZZAZIONE NUOVI PROG
    RAMMI
360 TT$="":Y$="
    "
370 V$(1)="[RVS]NOME PROGRAMMA : "
    :V$(2)="[RVS]ID DISCO   : "
380 FOR HG=1 TO 2:PRINT"[CLEAR]":P
    RINT"SCRITTURA NELLA MEMORIA"
390 PRINT"[HOME][4 DOWN]"V$(HG)
400 PRINT"[16 DOWN]"SPC(20)"[RVS]S
    CRIVERE <20 CAR[RVOFF]"
410 PRINT"[DOWN]"SPC(8)"F7 PER FIN
    IRE  -  F1 PER CERCARE"
420 PRINT"[HOME][7 DOWN]_"
430 GET A$:IF A$="" THEN 430
440 IF A$=CHR$(133) THEN GOTO 220
450 IF A$=CHR$(136) THEN PRINT"[CL
    EAR][5 DOWN]"SPC(18)"FINE":GOT
    O 700
460 IF A$=CHR$(13) THEN TT$=TT$+LE
    FT$(Z$+Y$,20):Z$="":NEXTHG:GOT
    O 510
470 Z$=Z$+A$:IF LEN(Z$)<20 THEN 50
    0
480 IF LEN(Z$)>19 THEN PRINT"[3 DO
    WN]  ( TITOLI = 16 CAR. ; ID
    = 2 CAR.)":STOP
490 Z$=CHR$(13):GOTO 460
500 PRINT"[HOME][7 DOWN]"Z$"_":GOT
    O 430
510 RESTORE :READ L:L=L+20
520 PRINT"[CLEAR]":PRINT"[HOME][3
    DOWN]"L"DATA"L","TT$
530 PRINT"1000 DATA"L
540 PRINT"GOTO350[HOME]"
550 POKE 198,3:FOR G=0 TO 3:POKE M
    M+G,13:NEXT
560 END
570 REM  ROUTINE SCRITTURA RICERCA
     EFFETTUATA
580 PRINT"RICERCA EFFETTUATA"
590 PRINT"[2 DOWN][RVS]NOME PROGRA
    MMA =  "MID$(X$,1,20)
600 PRINT"[2 DOWN][RVS]ID DISCO   =
      "MID$(X$,21,20)
610 IF BB=1 THEN PRINTSPC(19)"[6 D
    OWN][RVS]FINE DATI MEMORIZZATI
    "
620 PRINT"[5 DOWN]'RETURN' PER SEG
    UIRE  -  F7 PER FINIRE"
630 PRINT"[DOWN]"SPC(25)"F3 PER SC
    RIVERE"
640 GET P$:IF P$="" THEN 640
650 P=ASC(P$)
660 IF P=13GOTO 710
670 IF P=133GOTO 220
680 IF P=134GOTO 350
690 IF P=136 THEN PRINT"[CLEAR]"SP
    C(18)"FINE":END
700 PRINT"[5 DOWN](SAVE/REPLACE:AN
    TEPORRE AL TITOLO  @0: )":END
710 PRINT"[CLEAR]":RETURN
1000 DATA  1000
```

0. Nome *(facoltativo)* ____________

1. Indirizzo: ____________ via ____________

N. ________ città ____________

2. Età:

- ☐ Meno di 15 anni ☐ 15-18 anni
- ☐ 19-24 anni ☐ 25-30 anni
- ☐ 31-45 anni ☐ 46-65 anni
- ☐ 31-45 anni

3. Sesso:

☐ M ☐ F

4. Occupazione: ____________

5. Quale modello di computer possiedi?

- ☐ Vic 20 senza espansioni
- ☐ Vic 20 con espansione da 16K o oltre
- ☐ C 64
- ☐ C 16
- ☐ Altri modelli Commodore (specificare)
- ☐ Altre marche (specificare)
- ☐ Nessun computer.

6. Quali periferiche possiedi e/o intendi acquistare entro l'anno?

	Posseg.	**Intendo acquist.**
Joystick	☐	☐
Paddle	☐	☐
Tavoletta grafica	☐	☐
Penna ottica	☐	☐
Registratore a cassette	☐	☐
Floppy disk	☐	☐
Stampante	☐	☐
Plotter	☐	☐
Printer/plotter	☐	☐
Monitor	☐	☐

7. Come hai avuto questa copia di Commodore?

- ☐ L'ho comprata in edicola ☐ Sono abbonato
- ☐ Me l'ha passata un amico/familiare ☐ Altro

8. Da quanto tempo leggi Commodore?

	Regolar.	Saltuar.
Questa è la prima volta	☐	☐
Da qualche mese	☐	☐
Da oltre un anno	☐	☐
Fin dai primi mesi	☐	☐

9. Quali altre pubblicazioni leggi e apprezzi di più?

	Regolar.	Saltuar.
Applicando	☐	☐
Basic (enciclopedia)	☐	☐
Bit	☐	☐
Chip	☐	☐
Commodore	☐	☐
Commodore Computer Club	☐	☐
Computer	☐	☐
Computer & Electronics	☐	☐
Elettronica 2000	☐	☐
Elettronica & Computer	☐	☐
Enciclopedia dell'informatica	☐	☐
HC	☐	☐
Informatica Oggi	☐	☐
List	☐	☐
M & P Computer	☐	☐
Micro Top	☐	☐
MC Microcomputer	☐	☐
Nuova elettronica	☐	☐
Personal Time	☐	☐
PC Personal computer	☐	☐
Sinclair Computer	☐	☐
Sperimentare col computer	☐	☐
SuperSinc	☐	☐
SuperVic	☐	☐
Video giochi	☐	☐
Zerouno	☐	☐

10. Come valuti l'ultimo fascicolo di Commodore?

(dai un voto da 0 a 10 per ciascuna delle seguenti voci)

Livello professionale ____________

Varietà degli argomenti ____________

Chiarezza espositiva ____________

Approfondimento tecnico ____________

Tempestività ____________

Grafica/Impaginazione ____________

Pubblicità ____________

11. Cosa ti interessa leggere di più?

(indica solo le tre voci che per te contano di più)

- ☐ Articoli didattici sul Basic
- ☐ Articoli didattici sul linguaggio macchina
- ☐ Articoli didattici sull'hardware
- ☐ Recensioni del software in commercio
- ☐ Giochi brevi in basic
- ☐ Giochi brevi in LM
- ☐ Routine e programmi di utilità
- ☐ Applicazioni utili per la casa
- ☐ Applicazioni utili per l'ufficio
- ☐ Applicazioni utili per la scuola

Caro lettore, dicci chi sei

12. Come utilizzi il tuo computer?
(indicare solo le due applicazioni principali)
- ☐ A casa/ per giocare
- ☐ Per imparare a programmare
- ☐ Per il mio lavoro/il mio studio
- ☐ Quale strumento didattico
- ☐ Altro:

13. Quanti programmi software possiedi?
N. ____________________

Li hai comprati oppure te ne hanno regalato una copia?
- ☐ Comprati in un negozio
- ☐ Comprati in edicola
- ☐ Copiati
- ☐ Regalati

14. Quali riviste di programmi su cassetta sei solito comprare?

	Compro regolar.	Compro saltuar.
Commodore Club	☐	☐
Idea computer	☐	☐
Load'n Run	☐	☐
MondoSoft	☐	☐
Next	☐	☐
Open game	☐	☐
Peek	☐	☐
Play games	☐	☐
Play on tape	☐	☐
Poke	☐	☐
Run	☐	☐
16/48 Sinclair Computer	☐	☐
Soft	☐	☐
SuperSinc	☐	☐
SuperCommodore	☐	☐
Videoteca computer	☐	☐

15. Quante ore alla settimana dedichi al tuo computer?
☐ Quasi mai ☐ Meno di un'ora ☐ 1 - 5 ore
☐ 6 - 9 ore ☐ 10 - 19 ore ☐ 20 ore ed oltre

16. Quali altri hobby hai?
- ☐ Elettronica
- ☐ Modellismo
- ☐ Foto/cine
- ☐ Radiotrasmissioni
- ☐ Videoregistrazione
- ☐ Musica
- ☐ Collezionismo

17. Oltre te, quanti altri familiari o amici usano il tuo computer?
____

18. Oltre te, quanti altri familiari o amici leggono la tua copia di Commodore
____

19. A te capita leggere altre riviste di computer comprate da altri tuoi familiari o amici?
☐ si ☐ no

20. Puoi indicare i titoli principali? ____________________
____

PICCOLI ANNUNCI...
....
....
....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A

COMMODORE

V.le Famagosta, 75
20142 Milano